



PATENT

I hereby certify that on the date specified below, this correspondence is being deposited with the United States Postal Service as first-class mail in an envelope addressed to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants : Sapna George et al.
Application No. : 09/486,582
Filed : July 10, 2000
For : FAST SYNTHESIS SUB-BAND FILTERING METHOD FOR
DIGITAL SIGNAL DECODING

Examiner : Andrew C. Flanders
Art Unit : 2644
Docket No. : 851663.407
Date : June 20, 2006

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' BRIEF

Commissioner for Patents:

This brief is in furtherance of the Notice of Appeal, filed in this case December 20, 2005. The fees required under Section 1.17(c), and any required request for extension of time for filing this brief and fees therefor, are dealt with in the accompanying transmittal letter.

06/23/2006 TBESHAH1 00000005 09486582

580.00 OP

02 FC:1402

I. REAL PARTY IN INTEREST

The real party in interest is STMicroelectronics Asia Pacific (PTE) Limited, which is the assignee of the present invention. The assignment of record is to STMicroelectronics Asia

Best Available Copy

Pacific PTE Limited, having an address at 28 Ang Mo Kio Industrial Park 2, Singapore, 569508 Singapore.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals, interferences or judicial proceedings which directly affect or will be directly affected by or have a bearing on the Board's decision in this appeal. This application is a conversion of PCT Application No. PCT/SG97/00037 filed August 29, 1997, into a U.S. National Application.

III. STATUS OF CLAIMS

Claims 1-20 are currently pending in this application. All pending active claims are attached hereto as Appendix A.

Claims 1-6, 11, 14 and 16-19 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *de Sousa* (European Patent Application 0 564 089 A1) in view of *Uramoto* (European Patent Application 0 506 111 A2).

Claims 8-10 and 20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Uramoto* in view of *ISO Standard 11172-3*.

Claims 7, 12-13 and 15 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *de Sousa* in view of *Uramoto* and in further view of *ISO Standard 11172-3*.

The rejection of claims 1-20 is appealed.

IV. STATUS OF AMENDMENTS

No amendments were filed subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following summary discusses the subject matter of the independent appealed claims along with references to portions of the specification and drawings that provide support for the claims. The references are provided for exemplary purposes and are not intended to restrict the scope of the claims to the particular embodiments corresponding to the references provided.

Embodiments of the invention are directed to enhanced synthesis sub-band filtering during decoding of digital audio signals. Embodiments decode, for example, MPEG 1

layer 2 encoded audio. An Inverse Modified Discrete Cosine Transform (IMDCT) is implemented using addition/subtraction followed by multiplication.

Specifically, independent claim 1 is directed to a method of decoding digital audio data comprising the steps of obtaining an input sequence of data elements representing encoded audio samples, preprocessing the input sequence of data elements by calculating an array of sum data and an array of difference data using selected data elements from the input sequence, calculating a first sequence of output values using the array of sum data, calculating a second sequence of output values using the array of difference data and forming decoded audio signals from the first and second sequences of output values. Specification references are to the published PCT application. Support for independent claim 1 can be found in the last paragraph on page 2. More detail in the form of examples is provided on pages 2-12 and Figures 3-5.

Independent claim 8 is directed to a method of decoding a sequence of m, m an even positive integer, input digital audio data samples $S[k]$, where $k = 0, 1, \dots (m-1)$, to produce a set of n, n an even positive integer, output audio data samples $V[i]$, where $i = 0, 1, \dots (n-1)$, comprising the steps of:

- a) calculating an array of sum data $S_{ADD}[k]$ according to

$$S_{ADD}[k] = S[k] + S[m-1-k] \quad \text{for } k = 0, 1, \dots (m/2-1)$$
- b) calculating an array of difference data $S_{SUB}[k]$ according to

$$S_{SUB}[k] = S[k] - S[m-1-k] \quad \text{for } k = 0, 1, \dots (m/2-1)$$
- c) calculating a first output audio data sample by a multiply-accumulate

operation according to

$$V[2i] = V[2i] + N[i, k] * S_{ADD}[k] \quad \text{for } k = 0, 1, \dots (m/2-1)$$

$$\text{where } N[i, k] = \cos \left[\frac{(32 + 2i)(2k + 1)\pi}{64} \right]$$

- d) calculating a second output audio data sample by a multiply-accumulate

operation according to

$$V[2i+1] = V[2i+1] + N[i, k] * S_{SUB}[k] \quad \text{for } k = 0, 1, \dots (m/2-1)$$

$$\text{Where } N[1, k] = \cos \left[\frac{(32 + (2i + 1))(2k + 1)\pi}{64} \right]$$

e) and repeating steps c) and d) for $i = 0, 1, \dots (n/2-1)$ to obtain a full set of output data.

Support for claim 8 can be found in the last paragraph beginning on page 3 and continuing to page 4.

Independent claims 11 and 14 include means plus function elements. According to 37 CFR 41.37(c)(1)(v), such means plus function elements “must be identified and the structure, material, or acts described in the specification as corresponding to each claimed function must be set forth with reference to the specification by page and line number, and to the drawing, if any, by reference characters.” Accordingly, the following shows claims 11 and 14 together with the required information in parentheses.

11. A synthesis sub-band filter for use in decoding digital audio data, comprising:

means for receiving or retrieving an input sequence of data elements comprising encoded digital audio data; (**Page 4, first full paragraph; Fig. 2, input to audio decoder circuit 20 and the description thereof on page 5, first full paragraph; Figure 3, step 44; Figure 4, step 84; Figure 5, step 104**)

pre-calculation means for calculating an array of sum data and an array of difference data using selected data elements from the input sequence; (**Page 4, first full paragraph; Figure 2, audio decoder circuit 20 and the description thereof on page 5; Figure 3, step 52; Figure 4, step 86; Figure 5, steps 106, 108, 110**)

transform calculation means for calculating a first sequence of decoded output values using said array of sum data and a second sequence of decoded output values using said array of difference data (**Page 4, first full paragraph; Figure 2, audio decoder circuit 20 and the description thereof on page 5; Figure 3, step 52; Figure 4, steps 88, 90, 92; Figure 5, steps 112, 114, 116, 118, 120, 122**).

14. An MPEG decoder comprising:

means for receiving an input sequence of data elements comprising encoded digital audio data; **(Page 4, first full paragraph; Fig. 2, input to audio decoder circuit 20 and the description thereof on page 5, first full paragraph; Figure 3, step 44; Figure 4, step 84; Figure 5, step 104)**

means for calculating an array of sum data and an array of difference data using selected data elements from the input sequence; and **(Page 4, first full paragraph; Figure 2, audio decoder circuit 20 and the description thereof on page 5; Figure 3, step 52; Figure 4, step 86; Figure 5, steps 106, 108, 110)**

means for calculating a first sequence of decoded output values using said array of sum data and a second sequence of decoded output values using said array of difference data **(Page 4, first full paragraph; Figure 2, audio decoder circuit 20 and the description thereof on page 5; Figure 3, step 52; Figure 4, steps 88, 90, 92; Figure 5, steps 112, 114, 116, 118, 120, 122).**

VI. ISSUES

1. Whether claims 1-6, 11, 14 and 16-19 are unpatentable over *de Sousa* (European Patent Application 0 564 089 A1) in view of *Uramoto* (European Patent Application 0 506 111 A2).

2. Whether claims 8-10 and 20 are unpatentable over *Uramoto* in view of *ISO Standard 11172-3*.

3. Whether claims 7, 12-13 and 15 are unpatentable over *de Sousa* in view of *Uramoto* and in further view of *ISO Standard 11172-3*.

VII. ARGUMENT

A. *Introduction*

The Federal Circuit has held many times that the Examiner must provide objective evidence of a motivation for combining the teachings of cited references in the manner claimed. *E.g.*, *In re Sang-Su Lee*, 277 F.3d 1338, 1343; 61 USPQ2d 1430, 1433 (Fed. Cir. 2002). Further, “this factual question of motivation is material to patentability, and could not be resolved on subjective belief and unknown authority.” *Id.* at 277 F.3d 1343-1344; 61 USPQ2d 1433. Moreover, “the mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art suggested the desirability of the modification.” *In re Fritch*, 972 F.2d 1260, 1266; 23 USPQ2d 1780, 1783-84 (Fed. Cir. 1992).

The Examiner initially bears the burden of establishing a *prima facie* case of obviousness. *In re Bell*, 26 U.S.P.Q.2d 1529 (Fed. Cir. 1993); *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); *In re Piasecki*, 745 F.2d 1468, 1472, 223 U.S.P.Q. 785, 788 (Fed. Cir. 1984); MPEP § 2142. An Applicant may attack an obviousness rejection by showing that the Examiner has failed to properly establish a *prima facie* case or by presenting evidence tending to support a conclusion of non-obviousness. *In re Fritch*, 972 F.2d at 1265.

In order for an examiner to establish a *prima facie* case that an invention, as defined by a claim at issue, is obvious the examiner must: (1) show some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine the reference teachings; (2) there must be a reasonable expectation of success; and (3) the prior art reference (or the combined references) must teach or suggest all the claim limitations. MPEP § 2142. “The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant’s disclosure.” MPEP § 2143. The level of skill in the art cannot be relied upon to provide the suggestion to combine the references. MPEP § 2143.01 (citing *Al-Site Corp. v. VSI Int’l Inc.*, 174 F.3d 1308, 50 U.S.P.Q.2d 1161 (Fed. Cir. 1999)). The mere fact that the references can be combined or modified does not render the resultant

combination obvious unless the prior art also suggests the desirability of the combination. MPEP § 2143.01 (citing *In re Mills*, 916 F.2d 680, 16 U.S.P.Q. 2d 1430 (Fed. Cir. 1990)).

Moreover, a reference must be viewed as a whole, including portions that would lead away from the claimed invention. MPEP § 2141.03 (citing *W.L. Gore & Assoc., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 U.S.P.Q. 303 (Fed. Cir. 1983)). If the proposed modification would change the principles of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. MPEP § 2143.01 (citing *In re Ratti*, 270 F.2d 810, 123 U.S.P.Q. 349 (CCPA 1959)).

B. The Examiner Has Failed to Establish a Prima Facie Case of Obviousness

1. Claims 1-6, 11, 14 and 16-19 are not obvious over de Sousa in view of Uramoto

The Examiner rejected claims 1-6, 11, 14 and 16-19 under 35 U.S.C. § 103(a) as being unpatentable over de Sousa (European Patent Application 0 564 089 A1) in view of Uramoto (European Patent Application 0 506 111 A2). The Examiner has failed to establish a *prima facie* case of obviousness. Specifically, the Examiner has failed to show that the combination of de Sousa and Uramoto teaches, suggests or motivates the claimed invention. Moreover, Uramoto teaches away from the claimed invention.

Claim 1 recites, in part “[a] method of **decoding** digital audio data, comprising the steps of ... preprocessing the input sequence of data elements by calculating an array of sum data and an array of difference data **using selected data elements from the input sequence ...**” (emphasis added). Similarly, claim 11 recites, in part, “[a] synthesis sub-band filter for use in **decoding** digital audio data, comprising ... pre-calculation means for calculating an array of sum data and an array of difference data **using selected data elements from the input sequence ...**” (emphasis added). Claim 14 recites, in part, “[an] MPEG decoder comprising ... means for calculating an array of sum data and an array of difference data **using selected data elements from the input sequence ...**” (emphasis added).

The portions of de Sousa and Uramoto to which the Examiner points do not teach or suggest a method of decoding digital audio data, as recited. To the extent either addresses decoding, a different method is taught. Specifically, de Sousa does not address decoding except

in a conclusory form (*i.e.*, de Sousa in one paragraph notes that the “decoder has a very simple structure” but provides no details on the structure of the decoder or the methods employed). See Figure 12 of de Sousa and the terse description thereof on page 17, lines 34-39. Further, de Sousa does not disclose the way the processing of the input data is performed.

The portion of Uramoto to which the Examiner points teaches using the discrete cosine transform (DCT) for *encoding*. See Figure 5 of Uramoto and the accompanying description thereof on page 8, lines 15-37. Uramoto teaches using the inverse discrete cosine transform (IDCT) for decoding, which teaches post-processing “a sum and a difference between intermediate data.” In other words, intermediate multiplication of the input occurs and it is the intermediate data that is subjected to addition and subtraction. See, *e.g.*, the description of Figure 11 of Uramoto and the accompanying description thereof on page 10, line 48 through page 12, line 22. Accordingly, Uramoto teaches away from the claimed invention. Thus, the combination of de Sousa and Uramoto does not teach or suggest decoding digital audio data by “preprocessing the input sequence of data elements by calculating an array of sum data and an array of difference data ...; calculating a first sequence of output values using the array of sum data; calculating a second sequence of output values using the array of difference data; and forming decoded audio signals from the first and second sequences of output values” as recited. In fact, Uramoto teaches away from the claimed invention.

With regard to the decoding operation of Uramoto, the Examiner stated that Uramoto discloses a processing unit operable in a decoding application in which the processing unit is “in its same form as the processing unit disclosed in Fig. 5.” More specifically, and in reference to Fig. 11, Uramoto states “[p]ostprocessing section 7 has the same configuration as that of Fig. 5 or 6” (page 12, line 24). Although Uramoto discloses a postprocessing section 7 (Fig. 11) that has the same configuration as preprocessing section 1 (Figs. 4 or 5), postprocessing section 7 does not “calculat[e] an array of sum data and an array of difference data using selected data elements from the input sequence,” where the input sequence is an “input sequence of data elements representing encoded audio samples,” as claimed (emphasis added).

In reference to the postprocessing section 7 of the IDCT processor (Fig. 11) having the same configuration as the preprocessing section 1 of the DCT processor (Fig. 4), Uramoto states “input circuit 21 sequentially or alternately receives intermediate terms M_i ($i = 0$

to 3), N_i ($i = 0$ to 3) to apply a desired combination of the terms to adder/subtractors 22, 23 (or 26)” (page 12, lines 24-26). That is, the postprocessing section 7 operates on intermediate terms to generate output data x_i that is either a sum of intermediate terms (M_i and N_i) or a difference of intermediate terms, based upon the value of the integer i (page 12, lines 17-22). However, postprocessing section 7 does not operate on selected data elements from the input sequence to generate sum and difference data, where the selected data elements represent encoded audio samples. In other words, although postprocessing section 7 does operate as preprocessing section 1 to generate sum or difference data, postprocessing unit 7 does not generate an array of sum data and an array of difference data using selected data elements from the input sequence, as claimed.

Specifically, Uramoto discloses that $x_2 = M_2 + N_2 = A y_0 - C y_2 - A y_4 + B y_6 + F y_1 - D y_3 + G y_5 + E y_7$ (page 11, expression 13 and page 12, lines 7-20). That is, the sum output data generated by Uramoto (i.e., x_0 , x_1 , x_2 or x_3) is not comprised of “selected data elements from the input sequence,” as claimed. Instead, Uramoto generates an output x_2 , for example, that comprises additions and subtractions of products of input data (y_0 , y_1 , y_2 , y_3 , y_4 , y_5 , y_6 , y_7) and elements (A , B , C , D , E , F , G) of a coefficient matrix (expression 13, page 11).

The Examiner’s position appears to be that the combination of de Sousa and Uramoto *could* be further modified to achieve the claimed invention. The mere fact that references could be further modified is insufficient to establish obviousness, and the Examiner cites no motivation for this proposed further modification other than alleged skill in the art. Moreover, if the combination were further modified as the Examiner appears to suggest, the combination would not operate in accordance with the principles of operation of the decoder of Uramoto, which teaches an IDCT for decoding. Thus, the combination cannot be considered obvious.

Accordingly, de Sousa and Uramoto do not teach, suggest, or motivate, nor has the Examiner shown, decoding using “selected data elements from the input sequence” to generate either an array of sum data or an array of difference data, as claimed. Based at least upon the above arguments, Applicants respectfully submit that claims 1, 11 and 14 are not obvious over de Sousa in view of Uramoto.

Furthermore, since claims 2-6 and 18-19 depend either directly or indirectly from claim 1, and claims 16-17 depend from claim 14, Applicants submit that claims 2-6, 18-19 and claims 16-17 are allowable based at least upon the reasons given above in conjunction with claims 1, 11 and 14.

2. Claims 8-10 and 20 are not obvious over Uramoto in view of ISO Standard 11172-3

The Examiner rejected claims 8-10 and 20 under 35 U.S.C. § 103(a) as being unpatentable over Uramoto in view of ISO Standard 11172-3. The Examiner has failed to establish a prima facie case of obviousness.

Claim 8 recites: “[a] method of decoding ... input digital audio data samples ... comprising the steps of: ... calculating an array of sum data ... [;] calculating an array of difference data ... [;] calculating a first output audio data sample by a multiply-accumulate operation ... [;] calculating a second output audio data sample by a multiply-accumulate operation according.” The Examiner again points to the description of Figure 5 of Uramoto, which describes an *encoder*. As discussed above, Uramoto teaches away from the claimed invention by describing the use of a different method of decoding. See, e.g., the description of Figure 11 of Uramoto. Further, one would not be motivated to combine the inverse modified discrete cosine transform (IMDCT) with Uramoto, which as discussed above teaches the DCT for encoding and the IDCT for decoding.

Further, the sum output data $x_i = M_i + N_i$ for $i = 0, 1, 2, 3$ and the difference output data $x_i = M_i - N_i$ for $i = 4, 5, 6, 7$ generated by postprocessing section 7 (Uramoto, page 12, lines 20-22 and Fig. 11) is not the same as the array of sum data $SADD[k] = S[k] + S[m-1-k]$ and the array of difference data $SSUB[k] = S[k] - S[m-1-k]$ (for $k = 0, 1, \dots, (m/2-1)$), as claimed. Uramoto discloses M_i to be an intermediate term comprised of additions and/or subtractions of products of input data (y_0, y_2, y_4, y_6) with coefficients A, B and C, and N_i to be an intermediate term comprised of additions and/or subtractions of products of input data (y_1, y_3, y_5, y_7) with coefficients D, E, F and G (page 11, expression 13 to page 12, line 22). In contrast, $S[k]$ and $S[m-1-k]$ are coded input digital audio data samples. In other words, it is clear that $x_i = M_i - N_i$ does not equal either $SADD[k]$ or $SSUB[k]$, since $S[k]$ does not equal M_i and $S[m-1-k]$ does not equal N_i .

Based at least upon the above arguments, Applicants submit that claim 8 is not obvious over Uramoto in view of ISO Standard 11172-3. Furthermore, since claims 9-10 and 20 depend either directly or indirectly from claim 8, Applicants submit that claims 9-10 and 20 are allowable for at least the same reasons given above in conjunction with claim 8.

3. Claims 7, 12-13 and 15 are not obvious over de Sousa in view of Uramoto and ISO Standard 11172-3

The Examiner rejected claims 7, 12-13 and 15 under 35 U.S.C. § 103(a) as being unpatentable over de Sousa in view of Uramoto and in further view of ISO Standard 11172-3. The Examiner has failed to establish a prima facie case of obviousness.

Neither de Sousa nor ISO Standard 11172-3 remedy the deficiencies of Uramoto as discussed above in conjunction with claims 1, 11 and 14. Thus, Applicants respectfully submit that since claim 7 depends from claim 1, claims 12-13 depend either directly or indirectly from claim 11, and claim 15 depends from claim 14, claims 7, 12-13 and 15 are allowable based at least upon the reasons given above in conjunction with claims 1, 11 and 14, respectively, and request that claims 7, 12-13 and 15 be allowed.

VIII. CLAIMS APPENDIX

A copy of the claims as currently pending is attached hereto as Appendix A.

IX. EVIDENCE APPENDIX

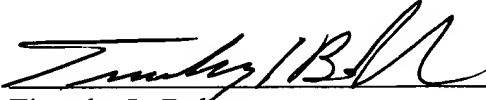
The Final Office Action mailed on June 20, 2005, and referred to herein is attached hereto as Appendix B.

A copy of the references cited by the Examiner and referred to herein is attached hereto as Appendix C. The references were cited in an Office Action mailed on June 20, 2005.

X. CONCLUSION

The Examiner has failed to establish a *prima facie* case that the claims are rendered unpatentable over *de Sousa*, whether considered alone or in combination with *Uramoto* and/or *ISO Standard 11172-3*. Moreover, *Uramoto* teaches away from the claims, and modifying *Uramoto* as suggested by the Examiner would change the principles of operation of *Uramoto*. Accordingly, allowance of the claims is respectfully requested.

Respectfully submitted,
Seed Intellectual Property Law Group PLLC



Timothy L. Boller
Registration No. 47,435

TLB:rr

Enclosures:

Postcard
Check No. 34751 for \$2,090
Transmittal Form
Fee Transmittal Form (+ copy)
Petition for Extension of Time
Appendix A
Appendix B
Appendix C

701 Fifth Avenue, Suite 6300
Seattle, Washington 98104-7092
Phone: (206) 622-4900
Fax: (206) 682-6031

747797_1.DOC

APPENDIX A
CLAIMS INVOLVED IN THE APPEAL

1. (Previously Presented) A method of decoding digital audio data, comprising the steps of:

obtaining an input sequence of data elements representing encoded audio samples;
preprocessing the input sequence of data elements by calculating an array of sum data and an array of difference data using selected data elements from the input sequence;

calculating a first sequence of output values using the array of sum data;
calculating a second sequence of output values using the array of difference data; and

forming decoded audio signals from the first and second sequences of output values.

2. (Previously Presented) A method as claimed in claim 1 wherein the array of sum data is obtained by adding together respective first and second data elements from the input sequence, the first and second data elements being selected from mutually exclusive sub-sequences of the input sequence.

3. (Previously Presented) A method as claimed in claim 1 wherein the array of difference data is obtained by subtracting respective first data elements from corresponding second data elements of the input sequence, the first and second data elements being selected from mutually exclusive sub-sequences of the input sequence.

4. (Previously Presented) A method as claimed in claim 1 wherein the step of calculating an array of sum data and an array of difference data comprises:

dividing the input data sequence into first and second equal sized sub-sequences, the first sub-sequence comprising the high order data elements of the input sequence and the second sub-sequence comprising the low order data elements of the input sequence;

calculating the array of sum data by adding together each respective data element of the first sub-sequence with a respective corresponding data element of the second sub-sequence; and

calculating the array of difference data by subtracting each respective data element of the first sub-sequence from a respective corresponding data element of the second sub-sequence.

5. (Previously Presented) A method as claimed in claim 1 wherein the step of calculating a first sequence of output values comprises performing a multiply-accumulate operation utilizing each of the sum data elements.

6. (Previously Presented) A method as claimed in claim 1, wherein the step of calculating a second sequence of output values comprises performing a multiply-accumulate operation utilizing each of the difference data elements.

7. (Previously Presented) A method as claimed in claim 1 wherein the input sequence of data elements is derived from MPEG encoded audio data, and wherein the decoded audio signals comprise pulse code modulation samples.

8. (Original) A method of decoding a sequence of m , m an even positive integer, input digital audio data samples $S[k]$, where $k = 0, 1, \dots (m-1)$, to produce a set of n , n an even positive integer, output audio data samples $V[i]$, where $i = 0, 1, \dots (n-1)$, comprising the steps of:

- c) calculating an array of sum data $S_{ADD}[k]$ according to
$$S_{ADD}[k] = S[k] + S[m-1-k] \quad \text{for } k = 0, 1, \dots (m/2-1)$$
- d) calculating an array of difference data $S_{SUB}[k]$ according to

$$S_{SUB}[k] = S[k] - S[m-1-k] \quad \text{for } k = 0, 1, \dots, (m/2-1)$$

c) calculating a first output audio data sample by a multiply-accumulate operation according to

$$V[2i] = V[2i] + N[i, k] * S_{ADD}[k] \quad \text{for } k = 0, 1, \dots, (m/2-1)$$

$$\text{where } N[i, k] = \cos \left[\frac{(32 + 2i)(2k + 1)\pi}{64} \right]$$

d) calculating a second output audio data sample by a multiply-accumulate operation according to

$$V[2i+1] = V[2i+1] + N[i, k] * S_{SUB}[k] \quad \text{for } k = 0, 1, \dots, (m/2-1)$$

$$\text{Where } N[1, k] = \cos \left[\frac{(32 + (2i + 1))(2k + 1)\pi}{64} \right]$$

e) and repeating steps c) and d) for $i = 0, 1, \dots, (n/2-1)$ to obtain a full set of output data.

9. (Original) A method as claimed in claim 8, wherein the number m of input digital audio data samples is 32, and the number n of output audio data samples is 32.

10. (Previously Presented) A method as claimed in claim 8 wherein the decoding steps are repeated for decoding a series of frames of encoded audio data in an MPEG format.

11. (Previously Presented) A synthesis sub-band filter for use in decoding digital audio data, comprising:

means for receiving or retrieving an input sequence of data elements comprising encoded digital audio data;

pre-calculation means for calculating an array of sum data and an array of difference data using selected data elements from the input sequence; and

transform calculation means for calculating a first sequence of decoded output values using said array of sum data and a second sequence of decoded output values using said array of difference data.

12. (Original) A synthesis sub-band filter as claimed in claim 11 wherein the pre-calculation means and transform calculation means collectively perform an inverse modified discrete cosine transform of the encoded digital audio data.

13. (Previously Presented) An MPEG decoder including a synthesis sub-band filter as claimed in claim 12.

14. (Previously Presented) An MPEG decoder comprising:
means for receiving an input sequence of data elements comprising encoded digital audio data;

means for calculating an array of sum data and an array of difference data using selected data elements from the input sequence; and

means for calculating a first sequence of decoded output values using said array of sum data and a second sequence of decoded output values using said array of difference data.

15. (Previously Presented) The MPEG decoder of claim 14 wherein the means for receiving an input sequence comprises a bitstream unpacking and decoding circuit.

16. (Previously Presented) The MPEG decoder of claim 14 wherein the means for calculating an array of sum data and an array of difference data comprises a reconstruction circuit.

17. (Previously Presented) The MPEG decoder of claim 14 wherein the means for calculating a first sequence of decoded output values comprises an inverse mapping circuit.

18. (Previously Presented) The method of claim 2 wherein the array of difference data is obtained by subtracting respective first data elements from corresponding second data elements of the input sequence, the first and second data elements being selected from mutually exclusive sub-sequences of the input sequence.

19. (Previously Presented) The method of claim 5 wherein the step of calculating a second sequence of output values comprises performing a multiply-accumulate operation utilizing each of the difference data elements.

20. (Previously Presented) The method of claim 9 wherein the decoding steps are repeated for decoding a series of frames of encoded audio data in an MPEG format.

APPENDIX B

FINAL OFFICE ACTION MAILED JUNE 20, 2005



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/486,582	07/10/2000	GEORGE SAPNA	851663.407 ✓	9626

7590 06/20/2005

DAVID V CARLSON
SEED INTELLECTUAL PROPERTY LAW GROUP
6300 COLUMBIA CENTER
701 5TH AVENUE
SEATTLE, WA 98104-7092

EXAMINER

FLANDERS, ANDREW C

ART UNIT

PAPER NUMBER

2644

RECEIVED

DATE MAILED: 06/20/2005

JUN 23 2005

SEED INTELLECTUAL PROPERTY
LAW GROUP PLLC

Please find below and/or attached an Office communication concerning this application or proceeding.

2 - month Response Due: AUG. 20, 20053 - month Response Due: SEPT. 20, 2005Notice of Appeal Due: DEC. 20, 2005

(6 - month period ends) Will Go Aban

(3 - month extension of the time required)

FINAL REJECTION

ENTERED IN DOCKET

RL

Office Action Summary

Application No.

09/486,582

Applicant(s)

SAPNA ET AL.

Examiner

Andrew C. Flanders

Art Unit

2644

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 24 May 2005.
- 2a) ☒ This action is FINAL. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 10 July 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____

- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

Response to Arguments

Applicant's arguments filed 24 May 2005 have been fully considered but they are not persuasive.

In regards to claims 1 and 11, Applicant states regarding the de Sousa art:

"de Sousa does not address decoding except in a conclusory form (i.e., de Sousa in one paragraph notes that the 'decoder has a very simple structure' but provides no details on the structure of the decoder or the methods employed)."

Examiner agrees with the above statements. However, the purpose of the addition of the Uramoto reference is to address the level of detail of the decoding. This is further referenced in the rejection of claim 1 in the previous action dated 24 February 2005 and acknowledged by the applicant within the arguments above.

Further, Applicant argues the portion of the Uramoto reference to which the previous action points to teaches using the DCT for encoding as opposed to decoding and that the method Uramoto teaches for decoding is different than applicants claimed invention.

However, in the previous reference the elements referred to in Uramoto are directed to a processing unit, regardless of its use in encoding or decoding. This processing unit is operable in a decoding application as is further evidenced by line 24 of page 12 in its same form as the processing unit disclosed in Fig. 5 to which the previous action refers to. As such, the processing section, as referenced in the previous action, may be used in the decoding of DCT encoded signals and reads upon

the claimed limitations of the application. Therefore the argument is not persuasive and the previous rejection stands.

In regards to claim 8, Applicant again states the portion of the Uramoto reference to which the previous action points to teaches using the DCT for encoding as opposed to decoding and that the method Uramoto teaches for decoding is different than applicants claimed invention.

For the same reasons as stated above regarding the arguments for claims 1 and 11 this argument is not persuasive.

Further applicant states one would not be motivated to combine the IMDCT with Uramoto. However, Examiner disagrees. Applicant has not given sufficient evidence explaining why one would not have been motivated to do so, applicant has only made a conclusory statement in which Uramoto teaches DCT and IDCT for encoding and decoding respectively. But, as stated in the previous rejection, using the IMDCT is one of the many implementations that one of ordinary skill in the art would have been motivated to use. As such the rejection stands.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1 – 6, 11, 14 and 16 - 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over de Sousa (European Patent Application 0 564 089 A1) in view of Uramoto (European Patent Application 0 506 111 A2).

Regarding **Claims 1, 11 and 14**, Sousa discloses a method of decoding digital audio data (page 2, lines 11-13), a step of obtaining an input sequence of data elements representing encoded audio samples (page 6 lines 51 – 53), preprocessing the input sequence of data elements (page 17 lines 32 – 39 and fig 12), performing a modified discrete cosine transform (abstract) and forming decoded audio signals (page 17 lines 32 – 39). Sousa does not disclose the way the processing of the input data is performed. Uramoto disclose a data processing method for video data (page 8, lines 15 – 37 and fig 5), method steps of calculating an array of sum data (page 8 lines 27 – 30), an array of difference data (page 8 line 31), calculating a first sequence of output values using the array of sum data (page 8 lines 32 – 37), calculating a second sequence of output values using the array of difference data (page 8 lines 32 – 37). It would have been obvious to one of ordinary skill in the art at the time of the invention namely when the same result is to be achieved; i.e. to reduce the amount of processing required for decoding, to apply the features of Uramoto to Sousa thereby arriving at a method according to claim 1.

Regarding **Claim 2**, in addition to the elements stated above regarding claim 1, Uramoto further discloses the input circuit receives sequentially output sets of data (x0,

Art Unit: 2644

x7) and then adder 22 adds the data i.e. $(x_0 + x_7)$ (page 8 lines 27 – 29) (i.e. wherein the array of sum data is obtained by adding together respective first and second data elements from the input sequence, the first and second data elements being selected from mutually exclusive sub-sequences of the input sequence).

Regarding **Claim 3**, in addition to the elements stated above regarding claim 1, Uramoto further disclose the input circuit receives sequentially output sets of data (x_0, x_7) and then subtractor 23 subtracts the data $(x_0 - x_7)$ (page 8 lines 27 – 31) (i.e. wherein the array of difference data is obtained by subtracting respective first data elements from corresponding second data elements of the input sequence, the first and second data elements being selected from mutually exclusive subsequences of the input sequence).

Regarding **Claim 4**, in addition to the elements stated above regarding claim 1, Uramoto discloses dividing X into (x_0, x_7) , the first and last elements of X (page 8 lines 27 – 28) (i.e. wherein the step of calculating an array of sum data and an array of difference data comprises dividing the input data sequence into first and second equal sized sub-sequences, the first sub-sequence comprising the higher order data elements of the input sequence and the second sub-sequence comprising the low order data elements of the input sequence), adder 22 adds the data i.e. $(x_0 + x_7)$ (page 8 lines 27 – 29) (i.e. calculating the array of sum data by adding together each respective data element of the first subsequence with a respective corresponding data element of the

Art Unit: 2644

second sub-sequence) and then subtractor 23 subtracts the data ($x_0 - x_7$) (page 8 lines 27 – 31) (i.e. and calculating the array of difference data by subtracting each respective data element of the first sub-sequence from a respective corresponding data element of the second sub-sequence):

Regarding **Claim 5**, in addition to the elements stated above regarding claim 1, the output of the addition and subtraction (fig. 5 element 500) is applied to a data rearranging circuit which supplies an output (fig. 7A elements 500 and 501), this output is then applied to a product sum operation circuit (fig. 8 element 501) (i.e. wherein the step of calculating a first sequence of output values comprises performing a multiply-accumulate operation utilizing each of the sum data elements).

Regarding **Claim 6**, in addition to the elements stated above regarding claim 1, the output of the addition and subtraction (fig. 5 element 500) is applied to a data rearranging circuit which supplies an output (fig. 7A elements 500 and 501), this output is then applied to a product sum operation circuit (fig. 8 element 501) (i.e. wherein the step of calculating a second sequence of output values comprises performing a multiply-accumulate operation utilizing each of the difference data elements).

Regarding **Claim 16**, in addition to the elements stated above regarding claim 14, Uramoto further discloses:

Art Unit: 2644

wherein the means for calculating an array of sum data and an array of difference data comprises a reconstruction circuit (i.e. the sum and difference operations are part of a processing circuit; pages 8 and 12).

Regarding **Claim 17**, in addition to the elements stated above regarding claim 14, Uramoto further discloses:

Wherein the means for calculating a first sequence of decoded output values comprises an inverse mapping circuit (i.e. the output circuit outputs the addition and subtraction data; page 8).

Regarding **Claim 18**, in addition to the elements stated above regarding claim 2, Uramoto further discloses:

wherein the array of difference data is obtained by subtracting respective first data elements from corresponding second data elements of the input sequence, the first and second data elements being selected from mutually exclusive subsequences of the input sequence (i.e. the subtractor selects the set of x_0 and x_7 to create a difference value from the sets of data of (x_0, x_7) , (x_1, x_6) , (x_2, x_5) and (x_3, x_4)).

Regarding **Claim 19**, in addition to the elements stated above regarding claim 1, the output of the addition and subtraction (fig. 5 element 500) is applied to a data rearranging circuit which supplies an output (fig 7A elements 500 and 501), this output is then applied to a product sum operation circuit (fig. 8 element 501) (i.e. wherein the

Art Unit: 2644

step of calculating a first sequence of output values comprises performing a multiply-accumulate operation utilizing each of the sum data elements).

Claims 8 – 10 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Uramoto (European Patent Application 0 506 111 A2) in view of ISO Standard 11172-3.

Regarding **Claim 8**, Uramoto discloses adder 22 adds the data i.e. $(x_0 + x_7)$ (page 8 lines 27 – 29) and subtractor 23 subtracts the data $(x_0 - x_7)$ (page 8 lines 27 – 31) i.e.

- a) calculating an array of sum data $S_{ADD}[k]$ according to
- $$S_{ADD}[k] = S[k] + S[m-1-k] \quad \text{for } k = 0, 1, \dots, (m/2-1)$$
- b) calculating an array of difference data $S_{SUB}[k]$ according to
- $$S_{SUB}[k] = S[k] - S[m-1-k] \quad \text{for } k = 0, 1, \dots, (m/2-1)$$

Uramoto does not disclose the rest of the claimed limitations in claim 8. ISO discloses an inverse modified discrete cosine transform (page 36). ISO also discloses multiplying samples by this function (page 41) i.e.

- c) calculating a first output audio data sample by a multiply-accumulate operation according to

$$V[2i] = V[2i] + N[i, k] * S_{ADD}[k] \quad \text{for } k = 0, 1, \dots, (m/2-1)$$

$$\text{where } N[i, k] = \cos \left[\frac{(32+2i)(2k+1)\pi}{64} \right]$$

- d) calculating a second output audio data sample by a multiply-accumulate operation according to

$$V[2i+1] = V[2i+1] + N[i, k] * S_{SUB}[k] \quad \text{for } k = 0, 1, \dots, (m/2-1)$$

$$\text{where } N[i, k] = \cos \left[\frac{(32+(2i+1))(2k+1)\pi}{64} \right]$$

- e) and repeating steps c) and d) for $i = 0, 1, \dots, (n/2-1)$ to obtain a full set of output data.

Art Unit: 2644

It would have been obvious to one of ordinary skill in the art at the time of the invention to use Uramoto's samples in ISO's decoder. It is merely one of many straightforward implementations of decoding audio within ISO's decoder and does not involve the exercise of inventive skill.

Regarding **Claim 9**, in addition to the elements stated above regarding claim 8, ISO discloses any number of samples from 12 – 36 (page 36).

Regarding **Claim 10**, in addition to the elements stated above regarding claim 8, ISO discloses decoding MPEG audio (page 41 and title).

Regarding **Claim 20**, in addition to the elements stated above regarding claim 9, wherein the steps of decoding are repeated for decoding a series of frames of encoded audio data in an MPEG format (i.e. the bit stream inputs a series of MPEG frames to be decoded; page 41).

Claims 7, 12, 13 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sousa (European Patent Application 0 564 089 A1) in view of Uramoto (European Patent Application 0 506 111 A2) and in further view of ISO Standard 11172-3.

Art Unit: 2644

Regarding **Claim 7**, in addition to the elements stated above regarding claim 1, the combination of de Sousa in view of Uramoto does not disclose the limitations of claim 7.

ISO discloses wherein the input sequence of data elements is derived from MPEG encoded audio data (page 41 and title), and wherein the decoded audio signals comprise pulse code modulation samples (i.e. the audio data left and right channel outputs; page 41). It would have been obvious to one of ordinary skill in the art at the time of the invention to use the samples of the combination in ISO's decoder. It is merely one of many straightforward implementations of decoding audio within ISO's decoder and does not involve the exercise of inventive skill.

Regarding **Claim 12**, in addition to the elements stated above regarding claim 11, ISO discloses the use of the inverse modified discrete cosine transform to decode audio data (pages 36 and 41).

Regarding **Claim 13**, in addition to the elements stated above regarding claim 11, ISO discloses decoding MPEG audio (page 41 and title).

Regarding **Claim 15**, in addition to the elements stated above regarding claim 14, ISO discloses wherein the means for receiving an input sequence comprises a bitstream unpacking and decoding circuit (page 41).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).


A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andrew C. Flanders whose telephone number is (571) 272-7516. The examiner can normally be reached on M-F 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Vivian Chin can be reached on (571) 272-7848. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2644

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

acf



VIVIAN CHIN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2600

Notice of References Cited	Application/Control No. 09/486,582	Applicant(s)/Patent Under Reexamination SAPNA ET AL.	
	Examiner Andrew C Flanders	Art Unit 2644	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N	0564089A1	06-1993		Sousa et al.	H04B 1/10
	O	0506111A2	09-1992		Uramoto et al.	G06F 15/332
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Interntaional Standard ISO/IEC 11172-3 01 August 1993
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

APPENDIX C

“A METHOD AND APPARATUS FOR THE PERCEPTUAL CODING OF AUDIO SIGNALS,” *Anibal Joao de Sousa Ferreira*, European Patent Application No. 0 564 089 A1, June 10, 1993

“DCT/IDCT PROCESSOR AND DATA PROCESSING METHOD,” *Shinichi Uramoto*, European Patent Application No. 0 506 111 A2, September 30, 1992

INTERNATIONAL STANDARD ISO/IEC 11172-3, August 1, 1993

CITED BY EXAMINER IN OFFICE ACTION MAILED JUNE 20, 2005



⑪ Publication number : **0 564 089 A1**

⑫

EUROPEAN PATENT APPLICATION

⑳ Application number : 93301433.4

㉔ Int. Cl.⁵ : **H04B 1/10, H03M 7/02**

㉕ Date of filing : 25.02.93

㉖ Priority : 02.03.92 US 844819

㉗ Date of publication of application :
06.10.93 Bulletin 93/40

㉘ Designated Contracting States :
DE FR GB IT NL

㉙ Applicant : **AMERICAN TELEPHONE AND
TELEGRAPH COMPANY**
32 Avenue of the Americas
New York, NY 10013-2412 (US)

㉚ Inventor : **de Sousa Ferrelra, Anibal Joao**
Av. Zeferino de Oliveira 239
P-4560 Penafiel (PT)
Inventor : **Johnston, James David**
8 Valley View Road
Warren, New Jersey 07059 (US)

㉛ Representative : **Buckley, Christopher Simon**
Thirsk et al
AT & T (UK) LTD. 5 Morningside Road
Woodford Green, Essex IG8 OTU (GB)

㉜ A method and apparatus for the perceptual coding of audio signals.

㉝ A method and apparatus for performing a Modified Discrete Cosine Transform on an audio signal is disclosed which utilizes a Discrete Fourier Transform. Illustratively, the MDCT spectral coefficients for the signal are generated from the real FFT spectral coefficients.

EP 0 564 089 A1

Best Available Copy

Cross-Reference to Related Applications and Materials

The following U.S. patent applications filed concurrently with the present application and assigned to the assignee of the present application are related to the present application and each is hereby incorporated herein as if set forth in its entirety: "RATE LOOP PROCESSOR FOR PERCEPTUAL ENCODER/DECODER, by J.D. Johnston; "A METHOD AND APPARATUS FOR CODING AUDIO SIGNALS BASED ON PERCEPTUAL MODEL," by J.D. Johnston; and "AN ENTROPY CODER," by J.D. Johnston and J.A. Reeds.

Field of the Invention

The present invention relates to processing of information signals, and more particularly, to the efficient encoding and decoding of monophonic and stereophonic audio signals, including signals representative of voice and music information, for storage or transmission.

Background of the Invention

Consumer, industrial, studio and laboratory products for storing, processing and communicating high quality audio signals are in great demand. For example, so-called compact disc ("CD") and digital audio tape ("DAT") recordings for music have largely replaced the long-popular phonograph record and cassette tape. Likewise, recently available digital audio tape ("DAT") recordings promise to provide greater flexibility and high storage density for high quality audio signals. See, also, Tan and Vermeulen, "Digital audio tape for data storage", *IEEE Spectrum*, pp. 34-38 (Oct. 1989). A demand is also arising for broadcast applications of digital technology that offer CD-like quality.

While these emerging digital techniques are capable of producing high quality signals, such performance is often achieved only at the expense of considerable data storage capacity or transmission bandwidth. Accordingly, much work has been done in an attempt to compress high quality audio signals for storage and transmission.

Most of the prior work directed to compressing signals for transmission and storage has sought to reduce the redundancies that the source of the signals places on the signal. Thus, such techniques as ADPCM, sub-band coding and transform coding described, e.g., in N.S. Jayant and P. Noll, "Digital Coding of Waveforms," Prentice-Hall, Inc. 1984, have sought to eliminate redundancies that otherwise would exist in the source signals.

In other approaches, the irrelevant information in source signals is sought to be eliminated using techniques based on models of the human perceptual system. Such techniques are described, e.g., in E.F. Schroeder and J.J. Platte, "MSC: Stereo Audio Coding with CD-Quality and 256 kBIT/SEC." *IEEE Trans. on Consumer Electronics*, Vol. CE-33, No. 4, November 1987; and Johnston, *Transform Coding of Audio Signals Using Noise Criteria*, Vol. 6, No. 2, *IEEE J.S.C.A.* (Feb. 1988).

Perceptual coding, as described, e.g., in the Johnston paper relates to a technique for lowering required bitrates (or reapportioning available bits) or total number of bits in representing audio signals. In this form of coding, a masking threshold for unwanted signals is identified as a function of frequency of the desired signal. Then, inter alia, the coarseness of quantizing used to represent a signal component of the desired signal is selected such that the quantizing noise introduced by the coding does not rise above the noise threshold, though it may be quite near this threshold. The introduced noise is therefore masked in the perception process. While traditional signal-to-noise ratios for such perceptually coded signals may be relatively low, the quality of these signals upon decoding, as perceived by a human listener, is nevertheless high.

Brandenburg et al, U.S. Patent 5,040,217, issued August 13, 1991, describes a system for efficiently coding and decoding high quality audio signals using such perceptual considerations. In particular, using a measure of the "noise-like" or "tone-like" quality of the input signals, the embodiments described in the latter system provides a very efficient coding for monophonic audio signals.

It is, of course, important that the coding techniques used to compress audio signals do not themselves introduce offensive components or artifacts. This is especially important when coding stereophonic audio information where coded information corresponding to one stereo channel, when decoded for reproduction, can interfere or interact with coding information corresponding to the other stereo channel. Implementation choices for coding two stereo channels include so-called "dual mono" coders using two independent coders operating at fixed bit rates. By contrast, "joint mono" coders use two monophonic coders but share one combined bit rate, i.e., the bit rate for the two coders is constrained to be less than or equal to a fixed rate, but trade-offs can be made between the bit rates for individual coders. "Joint stereo" coders are those that attempt to use inter-channel properties for the stereo pair for realizing additional coding gain.

It has been found that the independent coding of the two channels of a stereo pair, especially at low bit-rates, can lead to a number of undesirable psychoacoustic artifacts. Among them are those related to the localization of coding noise that does not match the localization of the dynamically imaged signal. Thus the human stereophonic perception process appears to add constraints to the encoding process if such mismatched localization is to be avoided. This finding is consistent with reports on binaural masking-level differences that appear to exist, at least for low frequencies, such that noise may be isolated spatially. Such binaural masking-level differences are considered to unmask a noise component that would be masked in a monophonic system. See, for example, B.C.J. Morre, "An Introduction to the Psychology of Hearing, Second Edition," especially chapter 5, Academic Press, Orlando, FL, 1982.

One technique for reducing psychoacoustic artifacts in the stereophonic context employs the ISO-WG11-MPEG-Audio Psychoacoustic II (ISO) Model. In this model, a second limit of signal-to-noise ratio ("SNR") is applied to signal-to-noise ratios inside the psychoacoustic model. However, such additional SNR constraints typically require the expenditure of additional channel capacity or (in storage applications) the use of additional storage capacity, at low frequencies, while also degrading the monophonic performance of the coding.

Summary of the Invention

Limitations of the prior art are overcome and a technical advance is made in a method and apparatus for coding a stereo pair of high quality audio channels in accordance with aspects of the present invention. Inter-channel redundancy and irrelevancy are exploited to achieve lower bit-rates while maintaining high quality reproduction after decoding. While particularly appropriate to stereophonic coding and decoding, the advantages of the present invention may also be realized in conventional dual monophonic stereo coders.

An illustrative embodiment of the present invention employs a filter bank architecture using a Modified Discrete Cosine Transform (MDCT). In order to code the full range of signals that may be presented to the system, the illustrative embodiment advantageously uses both L/R (Left and Right) and M/S (Sum/Difference) coding, switched in both frequency and time in a signal dependent fashion. A new stereophonic noise masking model advantageously detects and avoids binaural artifacts in the coded stereophonic signal. Interchannel redundancy is exploited to provide enhanced compression for without degrading audio quality.

The time behavior of both Right and Left audio channels is advantageously accurately monitored and the results used to control the temporal resolution of the coding process. Thus, in one aspect, an illustrative embodiment of the present invention, provides processing of input signals in terms of either a normal MDCT window, or, when signal conditions indicate, shorter windows. Further, dynamic switching between RIGHT/LEFT or SUM/DIFFERENCE coding modes is provided both in time and frequency to control unwanted binaural noise localization, to prevent the need for overcoding of SUM/DIFFERENCE signals, and to maximize the global coding gain.

A typical bitstream definition and rate control loop are described which provide useful flexibility in forming the coder output. Interchannel irrelevancies, are advantageously eliminated and stereophonic noise masking improved, thereby to achieve improved reproduced audio quality in jointly coded stereophonic pairs. The rate control method used in an illustrative embodiment uses an interpolation between absolute thresholds and masking threshold for signals below the rate-limit of the coder, and a threshold elevation strategy under rate-limited conditions.

In accordance with an overall coder/decoder system aspect of the present invention, it proves advantageously to employ an improved Huffman-like entropy coder/decoder to further reduce the channel bit rate requirements, or storage capacity for storage applications. The noiseless compression method illustratively used employs Huffman coding along with a frequency-partitioning scheme to efficiently code the frequency samples for L, R, M and S, as may be dictated by the perceptual threshold.

The present invention provides a mechanism for determining the scale factors to be used in quantizing the audio signal (i.e., the MDCT coefficients output from the analysis filter bank) by using an approach different from the prior art, and while avoiding many of the restrictions and costs of prior quantizer/rate-loops. The audio signals quantized pursuant to the present invention introduce less noise and encode into fewer bits than the prior art.

These results are obtained in an illustrative embodiment of the present invention whereby the utilized scale factor, is iteratively derived by interpolating between a scale factor derived from a calculated threshold of hearing at the frequency corresponding to the frequency of the respective spectral coefficient to be quantized and a scale factor derived from the absolute threshold of hearing at said frequency until the quantized spectral coefficients can be encoded within permissible limits.

Brief Description of the Drawings

- FIG. 1 presents an illustrative prior art audio communication/storage system of a type in which aspects of the present invention find application, and provides improvement and extension.
- FIG. 2 presents an illustrative perceptual audio coder (PAC) in which the advances and teachings of the present invention find application, and provide improvement and extension.
- FIG. 3 shows a representation of a useful masking level difference factor used in threshold calculations.
- FIG. 4 presents an illustrative analysis filter bank according to an aspect of the present invention.
- FIG. 5(a) through 5(e) illustrate the operation of various window functions.
- FIG. 6 is a flow chart illustrating window switching functionality.
- FIG. 7 is a block/flow diagram illustrating the overall processing of input signals to derive the output bitstream.
- FIG. 8 illustrates certain threshold variations.
- FIG. 9 is a flowchart representation of certain bit allocation functionality.
- FIG. 10 shows bitstream organization.
- FIGs 11a through 11c illustrate certain Huffman coding operations.
- FIG. 12 shows operations at a decoder that are complementary to those for an encoder.
- FIG. 13 is a flowchart illustrating certain quantization operations in accordance with an aspect of the present invention.
- FIG. 14(a) through 14(g) are illustrative windows for use with the filter bank of FIG. 4.

Detailed Description**1. Overview**

To simplify the present disclosure, the following patents, patent applications and publications are hereby incorporated by reference in the present disclosure as if fully set forth herein: U.S. Patent 5,040,217, issued August 13, 1991 by K. Brandenburg et al, United States Patent Application Serial No. 07/292,598, entitled *Perceptual Coding of Audio Signals*, filed December 30, 1988; J. D. Johnston, *Transform Coding of Audio Signals Using Perceptual Noise Criteria*, IEEE Journal on Selected Areas in Communications, Vol. 6, No. 2 (Feb. 1988); International Patent Application (PCT) WO 88/01811, filed March 10, 1988; United States Patent Application Serial No. 07/491,373, entitled *Hybrid Perceptual Coding*, filed March 9, 1990, Brandenburg et al, *Aspec: Adaptive Spectral Entropy Coding of High Quality Music Signals*, AES 90th Convention (1991); Johnston, J., *Estimation of Perceptual Entropy Using Noise Masking Criteria*, ICASSP, (1988); J. D. Johnston, *Perceptual Transform Coding of Wideband Stereo Signals*, ICASSP, (1989); E.F. Schroeder and J.J. Platta, "MSC: Stereo Audio Coding with CD-Quality and 256 kBIT/SEC," IEEE Trans. on Consumer Electronics, Vol. CE-33, No. 4, November 1987; and Johnston, *Transform Coding of Audio Signals Using Noise Criteria*, Vol. 6, No. 2, IEEE J.S.C.A. (Feb. 1988).

For clarity of explanation, the illustrative embodiment of the present invention is presented as comprising individual functional blocks (including functional blocks labeled as "processors"). The functions these blocks represent may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of executing software. (Use of the term "processor" should not be construed to refer exclusively to hardware capable of executing software.) Illustrative embodiments may comprise digital signal processor (DSP) hardware, such as the AT&T DSP16 or DSP32C, and software performing the operations discussed below. Very large scale integration (VLSI) hardware embodiments of the present invention, as well as hybrid DSP/VLSI embodiments, may also be provided.

FIG. 1 is an overall block diagram of a system useful for incorporating an illustrative embodiment of the present invention. At the level shown, the system of FIG. 1 illustrates systems known in the prior art, but modifications, and extensions described herein will make clear the contributions of the present invention. In FIG. 1, an analog audio signal 101 is fed into a preprocessor 102 where it is sampled (typically at 48 KHz) and converted into a digital pulse code modulation ("PCM") signal 103 (typically 16 bits) in standard fashion. The PCM signal 103 is fed into a perceptual audio coder 104 ("PAC") which compresses the PCM signal and outputs the compressed PAC signal to a communications channel/storage medium 105. From the communications channel/storage medium the compressed PAC signal is fed into a perceptual audio decoder 107 which decompresses the compressed PAC signal and outputs a PCM signal 108 which is representative of the compressed PAC signal. From the perceptual audio decoder, the PCM signal 108 is fed into a post-processor 109 which creates an analog representation of the PCM signal 108.

An illustrative embodiment of the perceptual audio coder 104 is shown in block diagram form in FIG. 2. As

in the case of the system illustrated in FIG. 1, the system of FIG. 2, without more, may equally describe certain prior art systems, e.g., the system disclosed in the Brandenburg, et al U.S. Patent 5,040,217. However, with the extensions and modifications described herein, important new results are obtained. The perceptual audio coder of FIG. 2 may advantageously be viewed as comprising an analysis filter bank 202, a perceptual model processor 204, a quantizer/rate-loop processor 206 and an entropy coder 208.

The filter bank 202 in FIG. 2 advantageously transforms an input audio signal in time/frequency in such manner as to provide both some measure of signal processing gain (i.e. redundancy extraction) and a mapping of the filter bank inputs in a way that is meaningful in light of the human perceptual system. Advantageously, the well-known Modified Discrete Cosine Transform (MDCT) described, e.g., in J.P. Princen and A.B. Bradley, "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation," IEEE Trans. ASSP, Vol. 34, No. 5, October, 1986, may be adapted to perform such transforming of the input signals.

Features of the MDCT that make it useful in the present context include its critical sampling characteristic, i.e. for every n samples into the filter bank, n samples are obtained from the filter bank. Additionally, the MDCT typically provides half-overlap, i.e. the transform length is exactly twice the length of the number of samples, n , shifted into the filterbank. The half-overlap provides a good method of dealing with the control of noise injected independently into each filter tap as well as providing a good analysis window frequency response. In addition, in the absence of quantization, the MDCT provides exact reconstruction of the input samples, subject only to a delay of an integral number of samples.

One aspect in which the MDCT is advantageously modified for use in connection with a highly efficient stereophonic audio coder is the provision of the ability to switch the length of the analysis window for signal sections which have strongly non-stationary components in such a fashion that it retains the critically sampled and exact reconstruction properties. The incorporated U.S. patent application by Ferreira and Johnston, entitled "A METHOD AND APPARATUS FOR THE PERCEPTUAL CODING OF AUDIO SIGNALS," (referred to hereinafter as the "filter bank application") filed of even date with this application, describes a filter bank appropriate for performing the functions of element 202 in FIG. 2.

The perceptual model processor 204 shown in FIG. 2 calculates an estimate of the perceptual importance, noise masking properties, or just noticeable noise floor of the various signal components in the analysis bank. Signals representative of these quantities are then provided to other system elements to provide improved control of the filtering operations and organizing of the data to be sent to the channel or storage medium. Rather than using the critical band by critical band analysis described in J.D. Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria," IEEE J. on Selected Areas in Communications, Feb. 1988, an illustrative embodiment of the present invention advantageously uses finer frequency resolution in the calculation of thresholds. Thus instead of using an overall tonality metric as in the last-cited Johnston paper, a tonality method based on that mentioned in K. Brandenburg and J.D. Johnston, "Second Generation Perceptual Audio Coding: The Hybrid Coder," AES 89th Convention, 1990 provides a tonality estimate that varies over frequency, thus providing a better fit for complex signals.

The psychoacoustic analysis performed in the perceptual model processor 204 provides a noise threshold for the L (Left), R (Right), M (Sum) and S (Difference) channels, as may be appropriate, for both the normal MDCT window and the shorter windows. Use of the shorter windows is advantageously controlled entirely by the psychoacoustic modal processor.

In operation, an illustrative embodiment of the perceptual model processor 204 evaluates thresholds for the left and right channels, denoted THR_L and THR_R . The two thresholds are then compared in each of the illustrative 35 coder frequency partitions (56 partitions in the case of an active window-switched block). In each partition where the two thresholds vary between left and right by less than some amount, typically 2dB, the coder is switched into M/S mode. That is, the left signal for that band of frequencies is replaced by $M = (L+R)/2$, and the right signal is replaced by $S = (L-R)/2$. The actual amount of difference that triggers the last-mentioned substitution will vary with bitrate constraints and other system parameters.

The same threshold calculation used for L and R thresholds is also used for M and S thresholds, with the threshold calculated on the actual M and S signals. First, the basic thresholds, denoted $BTHR_m$ and MLD_s , are calculated. Then, the following steps are used to calculate the stereo masking contribution of the M and S signals.

1. An additional factor is calculated for each of the M and S thresholds. This factor, called MLD_m and MLD_s , is calculated by multiplying the spread signal energy, (as derived, e.g., in J.D. Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria," IEEE J. on Selected Areas in Communications, Feb. 1988; K. Brandenburg and J.D. Johnston, "Second Generation Perceptual Audio Coding: The Hybrid Coder," AES 89th Convention, 1990; and Brandenburg, et al U.S. Patent 5,040,217) by a masking level difference factor shown illustratively in FIG. 3. This calculates a second level of detectability of noise across frequency in the M and S channels, based on the masking level differences shown in various sources.

2. The actual threshold for M (THR_m) is calculated as $THR_m = \max(BTHR_m, \min(BTHR_s, MLD_s))$ and the threshold $m = \max(BTHR_m, \min(BTHR_s, MLD_s))$ and the threshold for S is calculated as $THR_s = \max(BTHR_s, \min(BTHR_m, MLD_m))$.

In effect, the MLD signal substitutes for the BTHR signal in cases where there is a chance of stereo unmasking. It is not necessary to consider the issue of M and S threshold depression due to unequal L and R thresholds, because of the fact that L and R thresholds are known to be equal.

The quantizer and rate control processor 206 used in the illustrative coder of FIG. 2 takes the outputs from the analysis bank and the perceptual model, and allocates bits, noise, and controls other system parameters so as to meet the required bit rate for the given application. In some example coders this may consist of nothing more than quantization so that the just noticeable difference of the perceptual model is never exceeded, with no (explicit) attention to bit rate; in some coders this may be a complex set of iteration loops that adjusts distortion and bitrate in order to achieve a balance between bit rate and coding noise. A particularly useful quantizer and rate control processor is described in incorporated U.S. patent application by J.D. Johnston, entitled "RATE LOOP PROCESSOR FOR PERCEPTUAL ENCODER/DECODER," (hereinafter referred to as the "rate loop application") filed of even date with the present application. Also desirably performed by the rate loop processor 206, and described in the rate loop application, is the function of receiving information from the quantized analyzed signal and any requisite side information, inserting synchronization and framing information. Again, these same functions are broadly described in the incorporated Brandenburg, et al. U.S. patent 5,040,217.

Entropy coder 208 is used to achieve a further noiseless compression in cooperation with the rate control processor 206. In particular, entropy coder 208, in accordance with another aspect of the present invention, advantageously receives inputs including a quantized audio signal output from quantizer/rate-loop 206, performs a lossless encoding on the quantized audio signal, and outputs a compressed audio signal to the communications channel/storage medium 106.

Illustrative entropy coder 208 advantageously comprises a novel variation of the minimum-redundancy Huffman coding technique to encode each quantized audio signal. The Huffman codes are described, e.g., in D.A. Huffman, "A Method for the Construction of Minimum Redundancy Codes", *Proc. IRE*, 40:1098-1101 (1952) and T.M. Cover and J.A. Thomas, *Elements of Information Theory*, pp. 92-101 (1991). The useful adaptations of the Huffman codes advantageously used in the context of the coder of FIG. 2 are described in more detail in the incorporated U.S. patent application by J.D. Johnston and J. Reeds (hereinafter the "entropy coder application") filed of even date with the present application and assigned to the assignee of this application. Those skilled in the data communications arts will readily perceive how to implement alternative embodiments of entropy coder 208 using other noiseless data compression techniques, including the well-known Lempel-Ziv compression methods.

The use of each of the elements shown in FIG. 2 will be described in greater detail in the context of the overall system functionality; details of operation will be provided for the perceptual model processor 204.

2.1. The Analysis Filter Bank

The analysis filter bank 202 of the perceptual audio coder 104 receives as input pulse code modulated ("PCM") digital audio signals (typically 16-bit signals sampled at 48KHz), and outputs a representation of the input signal which identifies the individual frequency components of the input signal. Specifically, an output of the analysis filter bank 202 comprises a Modified Discrete Cosine Transform ("MDCT") of the input signal. See, J. Princen et al, "Sub-band Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation," *IEEE ICASSP*, pp. 2161-2164 (1987).

An illustrative analysis filter bank 202 according to one aspect of the present invention is presented in FIG. 4. Analysis filter bank 202 comprises an input signal buffer 302, a window multiplier 304, a window memory 306, an FFT processor 308, an MDCT processor 310, a concatenator 311, a delay memory 312 and a data selector 132.

The analysis filter bank 202 operates on frames. A frame is conveniently chosen as the 2N PCM input audio signal samples held by input signal buffer 302. As stated above, each PCM input audio signal sample is represented by M bits. Illustratively, $N = 512$ and $M = 16$.

Input signal buffer 302 comprises two sections: a first section comprising N samples in buffer locations 1 to N, and a second section comprising N samples in buffer locations N+1 to 2N. Each frame to be coded by the perceptual audio coder 104 is defined by shifting N consecutive samples of the input audio signal into the input signal buffer 302. Older samples are located at higher buffer locations than newer samples.

Assuming that, at a given time, the input signal buffer 302 contains a frame of 2N audio signal samples, the succeeding frame is obtained by (1) shifting the N audio signal samples in buffer locations 1 to N into buffer

locations N+1 to 2N, respectively, (the previous audio signal samples in locations N+1 to 2N may be either overwritten or deleted), and (2) by shifting into the input signal buffer 302, at buffer locations 1 to N, N new audio signal samples from preprocessor 102. Therefore, it can be seen that consecutive frames contain N samples in common: the first of the consecutive frames having the common samples in buffer locations 1 to N, and the second of the consecutive frames having the common samples in buffer locations N+1 to 2N. Analysis filter bank 202 is a critically sampled system (i.e., for every N audio signal samples received by the input signal buffer 302, the analysis filter bank 202 outputs a vector of N scalars to the quantizer/rate-loop 206).

Each frame of the input audio signal is provided to the window multiplier 304 by the input signal buffer 302 so that the window multiplier 304 may apply seven distinct data windows to the frame. Each data window is a vector of scalars called "coefficients". While all seven of the data windows have 2N coefficients (i.e., the same number as there are audio signal samples in the frame), four of the seven only have N/2 non-zero coefficients (i.e., one-fourth the number of audio signal samples in the frame). As is discussed below, the data window coefficients may be advantageously chosen to reduce the perceptual entropy of the output of the MDCT processor 310.

The information for the data window coefficients is stored in the window memory 306. The window memory 306 may illustratively comprise a random access memory ("RAM"), read only memory ("ROM"), or other magnetic or optical media. Drawings of seven illustrative data windows, as applied by window multiplier 304, are presented in FIG. 4. Typical vectors of coefficients for each of the seven data windows presented in FIG. 4 are presented in Appendix A. As may be seen in both FIG. 4 and in Appendix A, some of the data window coefficients may be equal to zero.

Keeping in mind that the data window is a vector of 2N scalars and that the audio signal frame is also a vector of 2N scalars, the data window coefficients are applied to the audio signal frame scalars through point-to-point multiplication (i.e., the first audio signal frame scalar is multiplied by the first data window coefficient, the second audio signal frame scalar is multiplied by the second data window coefficient, etc.). Window multiplier 304 may therefore comprise seven microprocessors operating in parallel, each performing 2N multiplications in order to apply one of the seven data window to the audio signal frame held by the input signal buffer 302. The output of the window multiplier 304 is seven vectors of 2N scalars to be referred to as "windowed frame vectors".

The seven windowed frame vectors are provided by window multiplier 304 to FFT processor 308. The FFT processor 308 performs an *odd-frequency* FFT on each of the seven windowed frame vectors. The odd-frequency FFT is an Discrete Fourier Transform evaluated at frequencies:

$$\frac{kf_M}{2N}$$

where $k = 1, 3, 5, \dots, 2N$, and f_M equals one half the sampling rate. The illustrative FFT processor 308 may comprise seven conventional decimation-in-time FFT processors operating in parallel, each operating on a different windowed frame vector. An output of the FFT processor 308 is seven vectors of 2N complex elements, to be referred to collectively as "FFT vectors".

FFT processor 308 provides the seven FFT vectors to both the perceptual model processor 204 and the MDCT processor 310. The perceptual model processor 204 uses the FFT vectors to direct the operation of the data selector 314 and the quantizer/rate-loop processor 208. Details regarding the operation of data selector 314 and perceptual model processor 204 are presented below.

MDCT processor 310 performs an MDCT based on the real components of each of the seven FFT vectors received from FFT processor 308. MDCT processor 310 may comprise seven microprocessors operating in parallel. Each such microprocessor determines one of the seven "MDCT vectors" of N real scalars based on one of the seven respective FFT vectors. For each FFT vector, $F(k)$, the resulting MDCT vector, $X(k)$, is formed as follows:

$$X(k) = \text{Re}[F(k)] \cos\left[\frac{\pi(2k+1)(1+N)}{4N}\right] \quad 1 \leq k \leq N.$$

The procedure need run k only to N , not $2N$, because of redundancy in the result. To wit, for $N < k \leq 2N$:

$$X(k) = -X(2N - k).$$

MDCT processor 310 provides the seven MDCT vectors to concatenator 311 and delay memory 312.

As discussed above with reference to window multiplier 304, four of the seven data windows have N/2 non-zero coefficients (see Figure 4c-f). This means that four of the windowed frame vectors contain only N/2 non-zero values. Therefore, the non-zero values of these four vectors may be concatenated into a single vector of length 2N by concatenator 311 upon output from MDCT processor 310. The resulting concatenation of these vectors is handled as a single vector for subsequent purposes. Thus, delay memory 312 is presented with four MDCT vectors, rather than seven.

Delay memory 312 receives the four MDCT vectors from MDCT processor 314 and concatenator 311 for

the purpose of providing temporary storage. Delay memory 312 provides a delay of one audio signal frame (as defined by input signal buffer 302) on the flow of the four MDCT vectors through the filter bank 202. The delay is provided by (i) storing the two most recent consecutive sets of MDCT vectors representing consecutive audio signal frames and (ii) presenting as input to data selector 314 the older of the consecutive sets of vectors. Delay memory 312 may comprise random access memory (RAM) of size:

$$M \times 2 \times 4 \times N$$

where 2 is the number of consecutive sets of vectors, 4 is the number of vectors in a set, N is the number of elements in an MDCT vector, and M is the number of bits used to represent an MDCT vector element.

Data selector 314 selects one of the four MDCT vectors provided by delay memory 312 to be output from the filter bank 202 to quantizer/rate-loop 206. As mentioned above, the perceptual model processor 204 directs the operation of data selector 314 based on the FFT vectors provided by the FFT processor 308. Due to the operation of delay memory 312, the seven FFT vectors provided to the perceptual model processor 204 and the four MDCT vectors concurrently provided to data selector 314 are *not* based on the same audio input frame, but rather on two consecutive input signal frames - the MDCT vectors based on the earlier of the frames, and the FFT vectors based on the later of the frames. Thus, the selection of a specific MDCT vector is based on information contained in the *next successive* audio signal frame. The criteria according to which the perceptual model processor 204 directs the selection of an MDCT vector is described in Section 2.2. *below*.

For purposes of an illustrative stereo embodiment, the above analysis filterbank 202 is provided for each of the left and right channels.

2.2. The Perceptual Model Processor

A perceptual coder achieves success in reducing the number of bits required to accurately represent high quality audio signals, in part, by introducing noise associated with quantization of information bearing signals, such as the MDCT information from the filter bank 202. The goal is, of course, to introduce this noise in an imperceptible or benign way. This noise shaping is primarily a frequency analysis instrument, so it is convenient to convert a signal into a spectral representation (e.g., the MDCT vectors provided by filter bank 202), compute the shape and amount of the noise that will be masked by these signals and injecting it by quantizing the spectral values. These and other basic operations are represented in the structure of the perceptual coder shown in FIG. 2.

The perceptual model processor 204 of the perceptual audio coder 104 illustratively receives its input from the analysis filter bank 202 which operates on successive *frames*. The perceptual model processor inputs then typically comprise seven Fast Fourier Transform (FFT) vectors from the analysis filter bank 202. These are the outputs of the FFT processor 308 in the form of seven vectors of $2N$ complex elements, each corresponding to one of the windowed frame vectors.

In order to mask the quantization noise by the signal, one must consider the spectral contents of the signal and the duration of a particular spectral pattern of the signal. These two aspects are related to masking in the frequency domain where signal and noise are approximately steady state - given the integration period of the hearing system - and also with masking in the time domain where signal and noise are subjected to different cochlear filters. The shape and length of these filters are frequency dependent.

Masking in the frequency domain is described by the concept of simultaneous masking. Masking in the time domain is characterized by the concept of premasking and postmasking. These concepts are extensively explained in the literature; see, for example, E. Zwicker and H. Fastl, *"Psychoacoustics, Facts and Models,"* Springer-Verlag, 1990. To make these concepts useful to perceptual coding, they are embodied in different ways.

Simultaneous masking is evaluated by using perceptual noise shaping models. Given the spectral contents of the signal and its description in terms of noise-like or tone-like behavior, these models produce an hypothetical masking threshold that rules the quantization level of each spectral component. This noise shaping represents the maximum amount of noise that may be introduced in the original signal without causing any perceptible difference. A measure called the *PERCEPTUAL ENTROPY (PE)* uses this hypothetical masking threshold to estimate the theoretical lower bound of the bitrate for transparent encoding. J. D. Johnston, *Estimation of Perceptual Entropy Using Noise Masking Criteria*, ICASSP, 1989.

Premasking characterizes the (in)audibility of a noise that starts some time before the masker signal which is louder than the noise. The noise amplitude must be more attenuated as the delay increases. This attenuation level is also frequency dependent. If the noise is the quantization noise attenuated by the first half of the synthesis window, experimental evidence indicates the maximum acceptable delay to be about 1 millisecond.

This problem is very sensitive and can conflict directly with achieving a good coding gain. Assuming stationary conditions - which is a false premiss - The coding gain is bigger for larger transforms, but, the quanti-

zation error spreads till the beginning of the reconstructed time segment. So, if a transform length of 1024 points is used, with a digital signal sampled at a rate of 48000Hz, the noise will appear at most 21 milliseconds before the signal. This scenario is particularly critical when the signal takes the form of a sharp transient in the time domain commonly known as an "attack". In this case the quantization noise is audible before the attack. The effect is known as pre-echo.

Thus, a fixed length filter bank is not a good perceptual solution nor a signal processing solution for non-stationary regions of the signal. It will be shown later that a possible way to circumvent this problem is to improve the temporal resolution of the coder by reducing the analysis/synthesis window length. This is implemented as a window switching mechanism when conditions of attack are detected. In this way, the coding gain achieved by using a long analysis/synthesis window will be affected only when such detection occurs with a consequent need to switch to a shorter analysis/synthesis window.

Postmasking characterizes the (in)audibility of a noise when it remains after the cessation of a stronger masker signal. In this case the acceptable delays are in the order of 20 milliseconds. Given that the bigger transformed time segment lasts 21 milliseconds (1024 samples), no special care is needed to handle this situation.

WINDOW SWITCHING

The PERCEPTUAL ENTROPY (PE) measure of a particular transform segment gives the theoretical lower bound of bits/sample to code that segment transparently. Due to its memory properties, which are related to premasking protection, this measure shows a significant increase of the PE value to its previous value -related with the previous segment- when some situations of strong non-stationarity of the signal (e.g. an attack) are presented. This important property is used to activate the window switching mechanism in order to reduce pre-echo. This window switching mechanism is not a new strategy, having been used, e.g., in the ASPEC coder, described in the ISO/MPEG Audio Coding Report, 1990, but the decision technique behind it is new using the PE information to accurately localize the non-stationarity and define the right moment to operate the switch.

Two basic window lengths: 1024 samples and 256 samples are used. The former corresponds to a segment duration of about 21 milliseconds and the latter to a segment duration of about 5 milliseconds. Short windows are associated in sets of 4 to represent as much spectral data as a large window (but they represent a "different" number of temporal samples). In order to make the transition from large to short windows and vice-versa it proves convenient to use two more types of windows. A START window makes the transition from large (regular) to short windows and a STOP window makes the opposite transition, as shown in FIG. 5b. See the above-cited Princen reference for useful information on this subject. Both windows are 1024 samples wide. They are useful to keep the system critically sampled and also to guarantee the time aliasing cancellation process in the transition region.

In order to exploit interchannel redundancy and irrelevancy, the same type of window is used for RIGHT and LEFT channels in each segment.

The stationarity behavior of the signal is monitored at two levels. First by large regular windows, then if necessary, by short windows. Accordingly, the PE of large (regular) window is calculated for every segment while the PE of short windows are calculated only when needed. However, the tonality information for both types is updated for every segment in order to follow the continuous variation of the signal.

Unless stated otherwise, a segment involves 1024 samples which is the length of a large regular window.

The diagram of FIG. 5a represents all the monitoring possibilities when the segment from the point $\frac{N}{2}$ till the point $\frac{3N}{2}$ is being analyzed. Related to diagram is the flowchart of FIG. 6 describes the monitoring sequence and decision technique. We need to keep in buffer three halves of a segment in order to be able to insert a START window prior to a sequence of short windows when necessary. FIGs. 5a-e explicitly considers the 50% overlap between successive segments.

The process begins by analysing a "new" segment with 512 new temporal samples (the remaining 512 samples belong to the previous segment). The PE of this new segment and the differential PE to the previous segment are calculated. If the latter value reaches a predefined threshold, then the existence of a non-stationarity inside the current segment is declared and details are obtained by processing four short windows with positions as represented in FIG. 5a. The PE value of each short window is calculated resulting in the ordered sequence: PE1, PE2, PE3 and PE4. From these values, the exact beginning of the strong non-stationarity of the signal is deduced. Only five locations are possible. They are identified in FIG. 4a as L1, L2, L3, L4 and L5.

As it will become evident, if the non-stationarity had occurred somewhere from the point $\frac{N}{2}$ till the point $\frac{15N}{16}$.

that situation would have been detected in the previous segment. It follows that the PE1 value does not contain relevant information about the stationarity of the current segment. The average PE of the short windows is compared with the PE of the large window of the same segment. A smaller PE reveals a more efficient coding situation. Thus if the former value is not smaller than the latter, then we assume that we are facing a degenerate situation and the window switching process is aborted.

It has been observed that for short windows the information about stationarity lies more on its PE value than on the differential to the PE value of the precedent window. Accordingly, the first window that has a PE value larger than a predefined threshold is detected. PE2 is identified with location L1, PE3 with L2 and PE4 with location L3. In either case, a START window is placed before the current segment that will be coded with short windows. A STOP window is needed to complete the process. There are, however, two possibilities. If the identified location where the strong non-stationarity of the signal begins is L1 or L2 then, this is well inside the short window sequence, no coding artifacts result and the coding sequence is depicted in FIG. 5b. If the location is L4, then, in the worst situation, the non-stationarity may begin very close to the right edge of the last short window. Previous results have consistently shown that placing a STOP window in coding conditions in these circumstances degrades significantly the reconstruction of the signal in this switching point. For this reason, another set of four short windows is placed before a STOP window. The resulting coding sequence is represented in FIG. 5a.

If none of the short PEs is above the threshold, the remaining possibilities are L4 or L5. In this case, the problem lies ahead of the scope of the short window sequence and the first segment in the buffer may be immediately coded using a regular large window.

To identify the correct location, another short window must be processed. It is represented in FIG. 5a by a dotted curve and its PE value, $PE_{1,n+1}$, is also computed. As it is easily recognized this short window already belongs to the next segment. If $PE_{1,n+1}$ is above the threshold, then, the location is L4 and, as depicted in FIG. 5c, a START window may be followed by a STOP window. In this case the spread of the quantization noise will be limited to the length of a short window, and a better coding gain is achieved. In the rare situation of the location being L5, then the coding is done according to the sequence of FIG. 5d. The way to prove that in this case that is right solution is by confirming that $PE_{2,n+1}$ will be above the threshold. $PE_{2,n+1}$ is the PE of the short window (not represented in FIG. 5) immediately following the window identified with $PE_{1,n+1}$.

As mentioned before for each segment, RIGHT and LEFT channels use the same type of analysis/synthesis window. This means that a switch is done for both channels when at least one channel requires it.

It has been observed that for low bitrate applications the solution of FIG. 5c, although representing a good local psychoacoustic solution, demands an unreasonably large number of bits that may adversely affect the coding quality of subsequent segments. For this reason, that coding solution may eventually be inhibited.

It is also evident that the details of the reconstructed signal when short windows are used are closer to the original signal than when only regular large window are used. This is so because the attack is basically a wide bandwidth signal and may only be considered stationary for very short periods of time. Since short windows have a greater temporal resolution than large windows, they are able to follow and reproduce with more fidelity the varying pattern of the spectrum. In other words, this is the difference between a more precise local (in time) quantization of the signal and a global (in frequency) quantization of the signal.

The final masking threshold of the stereophonic coder is calculated using a combination of monophonic and stereophonic thresholds. While the monophonic threshold is computed independently for each channel, the stereophonic one considers both channels.

The independent masking threshold for the RIGHT or the LEFT channel is computed using a psychoacoustic model that includes an expression for tone masking noise and noise masking tone. The latter is used as a conservative approximation for a noise masking noise expression. The monophonic threshold is calculated using the same procedure as previous work. In particular, a tonality measure considers the evolution of the power and the phase of each frequency coefficient across the last three segments to identify the signal as being more tone-like or noise-like. Accordingly, each psychoacoustic expression is more or less weighted than the other. These expressions found in the literature were updated for better performance. They are defined as:

$$TMN_{db} = 19.5 + \text{bark} \frac{18.0}{26.0}$$

$$NMT_{db} = 6.56 - \text{bark} \frac{3.06}{26.0}$$

where *bark* is the frequency in Bark scale. This scale is related to what we may call the *cochlear filters* or *critical bands* which, in turn, are identified with constant length segments of the basilar membrane. The final threshold is adjusted to consider absolute thresholds of masking and also to consider a partial premasking protection.

A brief description of the complete monophonic threshold calculation follows. Some terminology must be introduced in order to simplify the description of the operations involved.

The spectrum of each segment is organized in three different ways, each one following a different purpose.

5 1. First, it may be organized in partitions. Each partition has associated one single Bark value. These partitions provide a resolution of approximately either one MDCT line or 1/3 of a critical band, whichever is wider. At low frequencies a single line of the MDCT will constitute a coder partition. At high frequencies, many lines will be combined into one coder partition. In this case the Bark value associated is the median Bark point of the partition. This partitioning of the spectrum is necessary to insure an acceptable resolution
10 for the spreading function. As will be shown later, this function represents the masking influence among neighboring critical bands.

2. Secondly, the spectrum may be organized in bands. Bands are defined by a parameter file. Each band groups a number of spectral lines that are associated with a single scale factor that results from the final masking threshold vector.

15 3. Finally, the spectrum may also be organized in sections. It will be shown later that sections involve an integer number of bands and represent a region of the spectrum coded with the same Huffman code book. Three indices for data values are used. These are:

$\omega \rightarrow$ indicates that the calculation is indexed by frequency in the MDCT line domain.

20 $b \rightarrow$ indicates that the calculation is indexed in the threshold calculation partition domain. In the case where we do a convolution or sum in that domain, bb will be used as the summation variable.

$n \rightarrow$ indicates that the calculation is indexed in the coder band domain.

Additionally some symbols are also used:

1. The index of the calculation partition, b .
2. The lowest frequency line in the partition, ω_{low_b} .
- 25 3. The highest frequency line in the partition, ω_{high_b} .
4. The median bark value of the partition, $bval_b$.
5. The value for tone masking noise (in dB) for the partition, TMN_b .
6. The value for noise masking tone (in dB) for the partition, NMT_b .

30 Several points in the following description refer to the "spreading function". It is calculated by the following method:

$$tmpx = 1.05(j - i).$$

Where i is the bark value of the signal being spread, j the bark value of the band being spread into, and $tmpx$ is a temporary variable.

$$x = 8 \text{ minimum}((tmpx - .5)^2 - 2(tmpx - .5), 0)$$

35 Where x is a temporary variable, and $\text{minimum}(a,b)$ is a function returning the more negative of a or b .

$$tmpy = 15.811389 + 7.5(tmpx + .474) - 17.5(1 + (tmpx + .474)^2)^.5$$

where $tmpy$ is another temporary variable.

if ($tmpy < -100$) then $\{sprdngf(i,j) = 0\}$ else $\{sprdngf(i,j) = 10^{\frac{(x + tmpy)}{10}}\}$.

40 Steps In Threshold Calculation

The following steps are the necessary steps for calculation the SMR_n used in the coder.

1. Concatenate 512 new samples of the input signal to form another 1024 samples segment. Please refer to FIG. 5a.

45 2. Calculate the complex spectrum of the input signal using the O-FFT as described in 2.0 and using a sine window.

3. Calculate a predicted r and ϕ

The polar representation of the transform is calculated. r_n and ϕ_n represent the magnitude and phase components of a spectral line of the transformed segment.

50 A predicted magnitude, \hat{r}_n , and phase, $\hat{\phi}_n$, are calculated from the preceding two threshold calculation blocks' r and ϕ :

$$\begin{aligned}\hat{r}_n &= 2r_n(t-1) - r_n(t-2) \\ \hat{\phi}_n &= 2\phi_n(t-1) - \phi_n(t-2)\end{aligned}$$

55 where t represents the current block number, $t-1$ indexes the previous block's data, and $t-2$ indexes the data from the threshold calculation block before that.

4. Calculate the unpredictability measure c_n

c_n , the unpredictability measure, is:

$$c_{\omega} = \frac{((r_{\omega} \cos \phi_{\omega} - \hat{r}_{\omega} \cos \hat{\phi}_{\omega})^2 + (r_{\omega} \sin \phi_{\omega} - \hat{r}_{\omega} \sin \hat{\phi}_{\omega})^2)^{.5}}{r_{\omega} + \text{abs}(\hat{r}_{\omega})}$$

5. Calculate the energy and unpredictability in the threshold calculation partitions.
The energy in each partition, e_b , is:

$$e_b = \sum_{\omega=\omega_{low_b}}^{\omega_{high_b}} r_{\omega}^2$$

and the weighted unpredictability, c_b , is:

$$c_b = \sum_{\omega=\omega_{low_b}}^{\omega_{high_b}} r_{\omega}^2 c_{\omega}$$

6. Convolve the partitioned energy and unpredictability with the spreading function.

$$ecb_b = \sum_{bb=1}^{bmax} e_{bb} \text{sprdngf}(bval_{bb}, bval_b)$$

$$ct_b = \sum_{bb=1}^{bmax} c_{bb} \text{sprdngf}(bval_{bb}, bval_b)$$

Because ct_b is weighted by the signal energy, it must be renormalized to cb_b .

$$cb_b = \frac{ct_b}{ecb_b}$$

At the same time, due to the non-normalized nature of the spreading function, ecb_b should be renormalized and the normalized energy en_b calculated.

$$en_b = \frac{ecb_b}{rnorm_b}$$

The normalization coefficient, $rnorm_b$, is:

$$rnorm_b = \frac{1}{\sum_{bb=0}^{bmax} \text{sprdngf}(bval_{bb}, bval_b)}$$

7. Convert cb_b to tb_b .

$$tb_b = -.299 - .43 \log_{10}(cb_b)$$

Each tb_b is limited to the range of $0 \leq tb_b \leq 1$.

8. Calculate the required SNR in each partition.

$$TMN_b = 19.5 + bval_b \frac{18.0}{26.0}$$

$$NMT_b = 6.56 - bval_b \frac{3.06}{26.0}$$

Where TMN_b is the tone masking noise in dB and NMT_b is the noise masking tone value in dB.

The required signal to noise ratio, SNR_b , is:

$$SNR_b = tb_b TMN_b + (1 - tb_b) NMT_b$$

9. Calculate the power ratio.

The power ratio, bc_b , is:

$$bc_b = 10^{\frac{-SMR_b}{10}}$$

10. Calculation of actual energy threshold, nb_b .

$$nb_b = en_b bc_b$$

11. Spread the threshold energy over MDCT lines, yielding nb_w .

$$nb_w = \frac{nb_b}{\omega_{high_b} - \omega_{low_b} + 1}$$

12. Include absolute thresholds, yielding the final energy threshold of audibility, thr_w .

$$thr_w = \max(nb_w, absth_r).$$

The dB values of $absth_r$ shown in the "Absolute Threshold Tables" are relative to the level that a sine wave of $\pm \frac{1}{2}$ lsb has in the MDCT used for threshold calculation.

The dB values must be converted into the energy domain after considering the MDCT normalization actually used.

13. Pre-echo control

14. Calculate the signal to mask ratios, SMR_n .

The table of "Bands of the Coder" shows

1. The index, n , of the band.

2. The upper index, ω_{high_n} of the band n . The lower index, ω_{low_n} , is computed from the previous band as $\omega_{high_{n-1}} + 1$.

To further classify each band, another variable is created. The width index, $width_n$, will assume a value $width_n = 1$ if n is a perceptually narrow band, and $width_n = 0$ if n is a perceptually wide band. The former case occurs if

$$bval_{\omega_{high_n}} - bval_{\omega_{low_n}} < \text{bandlength}$$

bandlength is a parameter set in the initialization routine. Otherwise the latter case is assumed.

Then, if ($width_n = 1$), the noise level in the coder band, $nband_n$, is calculated as:

$$nband_n = \frac{\sum_{\omega=\omega_{low_n}}^{\omega_{high_n}} thr_{\omega}}{\omega_{high_n} - \omega_{low_n} + 1}$$

else,

$$nband_n = \text{minimum}(thr_{\omega_{low_n}}, \dots, thr_{\omega_{high_n}})$$

Where, in this case, $\text{minimum}(a, \dots, z)$ is a function returning the most negative or smallest positive argument of the arguments a, \dots, z .

The ratios to be sent to the decoder, SMR_n , are calculated as

$$SMR_n = 10 \log_{10} \left(\frac{(12.0 \cdot nband_n)^{1.5}}{\text{minimum}(absth_r)} \right)$$

It is important to emphasize that since the tonality measure is the output of a spectrum analysis process, the analysis window has a sine form for all the cases of large or short segments. In particular, when a segment is chosen to be coded as a START or STOP window, its tonality information is obtained considering a sine window; the remaining operations, e.g. the threshold calculation and the quantization of the coefficients, consider the spectrum obtained with the appropriate window.

STEREOPHONIC THRESHOLD

The stereophonic threshold has several goals. It is known that most of the time the two channels sound "alike". Thus, some correlation exists that may be converted in coding gain. Looking into the temporal representation of the two channels, this correlation is not obvious. However, the spectral representation has a number of interesting features that may advantageously be exploited. In fact, a very practical and useful possibility is to create a new basis to represent the two channels. This basis involves two orthogonal vectors, the vector SUM and the vector DIFFERENCE defined by the following linear combination:

$$\begin{bmatrix} \text{SUM} \\ \text{DIF} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \text{RIGHT} \\ \text{LEFT} \end{bmatrix}$$

These vectors, which have the length of the window being used, are generated in the frequency domain since the transform process is by definition a linear operation. This has the advantage of simplifying the computational load.

The first goal is to have a more decorrelated representation of the two signals. The concentration of most of the energy in one of these new channels is a consequence of the redundancy that exists between RIGHT and LEFT channels and on average, leads always to a coding gain.

A second goal is to correlate the quantization noise of the RIGHT and LEFT channels and control the localization of the noise or the *unmasking effect*. This problem arises if RIGHT and LEFT channels are quantized and coded independently. This concept is exemplified by the following context: supposing that the threshold of masking for a particular signal has been calculated, two situations may be created. First we add to the signal an amount of noise that corresponds to the threshold. If we present this same signal with this same noise to the two ears then the noise is masked. However, if we add an amount of noise that corresponds to the threshold to the signal and present this combination to one ear; do the same operation for the other ear but with noise uncorrelated with the previous one, then the noise is not masked. In order to achieve masking again, the noise at both ears must be reduced by a level given by the masking level differences (MLD).

The unmasking problem may be generalized to the following form: the quantization noise is not masked if it does not follow the localization of the masking signal. Hence, in particular, we may have two limit cases: center localization of the signal with unmasking more noticeable on the sides of the listener and side localization of the signal with unmasking more noticeable on the center line.

The new vectors SUM and DIFFERENCE are very convenient because they express the signal localized on the center and also on both sides of the listener. Also, they enable to control the quantization noise with center and side image. Thus, the unmasking problem is solved by controlling the protection level for the MLD through these vectors. Based on some psychoacoustic information and other experiments and results, the MLD protection is particularly critical for very low frequencies to about 3KHz. It appears to depend only on the signal power and not on its tonality properties. The following expression for the MLD proved to give good results:

$$\text{MLD}_{ab}(l) = 25.5 \left\{ \cos \frac{\pi b(l)}{32.0} \right\}^2$$

where l is the partition index of the spectrum (see [7]), and $b(l)$ is the *bark*-frequency of the center of the partition l . This expression is only valid for $b(l) \leq 16.0$ i.e. for frequencies below 3KHz. The expression for the MLD threshold is given by:

$$\text{THR}_{\text{MLD}}(l) = C(l) 10^{-\frac{\text{MLD}_{ab}(l)}{10}}$$

$C(l)$ is the spread signal energy on the basilar membrane, corresponding only to the partition l .

A third and last goal is to take advantage of a particular stereophonic signal image to extract irrelevance from *directions* of the signal that are masked by that image. In principle, this is done only when the stereo image is strongly defined in one direction, in order to not compromise the richness of the stereo signal. Based on the vectors SUM and DIFFERENCE, this goal is implemented by postulating the following two dual principles:

1. If there is a strong depression of the signal (and hence of the noise) on both sides of the listener, then an increase of the noise on the middle line (center image) is perceptually tolerated. The upper bound is the side noise.

2. If there is a strong localization of the signal (and hence of the noise) on the middle line, then an increase of the (correlated) noise on both sides is perceptually tolerated. The upper bound is the center noise. However, any increase of the noise level must be corrected by the MLD threshold.

According to these goals, the final stereophonic threshold is computed as follows. First, the thresholds for channels SUM and DIFFERENCE are calculated using the monophonic models for noise-masking-tone and tone-masking-noise. The procedure is exactly the one presented in 3.2 till step 10. At this point we have the actual energy threshold per band, en_b , for both channels. By convenience, we call them $\text{THR}_{n_{\text{SUM}}}$ and $\text{THR}_{n_{\text{DIF}}}$, respectively for the channel SUM and the channel DIFFERENCE.

Secondly, the MLD threshold for both channels i.e. $\text{THR}_{n_{\text{MLS,SUM}}}$ and $\text{THR}_{n_{\text{MLD,DIF}}}$ are also calculated by:

$$\text{THR}_{n_{\text{MLD,SUM}}} = en_{b,\text{SUM}} 10^{-\frac{\text{MLD}_{ab}}{10}}$$

$$THR_{n_{MLD,DIF}} = en_{b,DIF} 10 \cdot \frac{MLD_{n_b}}{10}$$

The MLD protection and the stereo irrelevance are considered by computing:

$$n_{thr_{SUM}} = \text{MAX}[THR_{n_{SUM}}, \text{MIN}(THR_{n_{DIF}}, THR_{n_{MLD,DIF}})]$$

$$n_{thr_{DIF}} = \text{MAX}[THR_{n_{DIF}}, \text{MIN}(THR_{n_{SUM}}, THR_{n_{MLD,SUM}})]$$

After these operations, the remaining steps after the 11th, as presented in 3.2 are also taken for both channels. In essence, these last thresholds are further adjusted to consider the absolute threshold and also a partial premasking protection. It must be noticed that this premasking protection was simply adopted from the monophonic case. It considers a monaural time resolution of about 2 milliseconds. However, the binaural time resolution is as accurate as 6 microseconds! To conveniently code stereo signals with relevant stereo image based on interchannel time differences, is a subject that needs further investigation.

STEREOPHONIC CODER

The simplified structure of the stereophonic coder is presented in FIG. 12. For each segment of data being analyzed, detailed information about the independent and relative behavior of both signal channels may be available through the information given by large and short transforms. This information is used according to the necessary number of steps needed to code a particular segment. These steps involve essentially the selection of the analysis window, the definition on a band basis of the coding mode (R/L or S/D), the quantization and Huffman coding of the coefficients and scale factors and finally, the bitstream composing

Coding Mode Selection

When a new segment is read, the tonality updating for large and short analysis windows is done. Monophonic thresholds and the PE values are calculated according to the technique described in Section 3.1. This gives the first decision about the type of window to be used for both channels.

Once the window sequence is chosen, an orthogonal coding decision is then considered. It involves the choice between independent coding of the channels, mode RIGHT/LEFT (R/L) or joint coding using the SUM and DIFFERENCE channels (S/D). This decision is taken on a band basis of the coder. This is based on the assumption that the binaural perception is a function of the output of the same critical bands at the two ears. If the threshold at the two channels is very different, then there is no need for MLD protection and the signals will not be more decorrelated if the channels SUM and DIFFERENCE are considered. If the signals are such that they generate a stereo image, then a MLD protection must be activated and additional gains may be exploited by choosing the S/D coding mode. A convenient way to detect this latter situation is by comparing the monophonic threshold between RIGHT and LEFT channels. If the thresholds in a particular band do not differ by more than a predefined value, e.g. 2dB, then the S/D coding mode is chosen. Otherwise the independent mode R/L is assumed. Associated with each band is a one bit flag that specifies the coding mode of that band and that must be transmitted to the decoder as side information. >From now on it is called a coding mode flag.

The coding mode decision is adaptive in time since for the same band it may differ for subsequent segments, and is also adaptive in frequency since for the same segment, the coding mode for subsequent bands may be different. An illustration of a coding decision is given in FIG. 13. This illustration is valid for long and also short segments.

At this point it is clear that since the window switching mechanism involves only monophonic measures, the maximum number of PE measures per segment is 10 (2 channels * [1 large window + 4 short windows]). However, the maximum number of thresholds that we may need to compute per segment is 20 and therefore 20 tonality measures must be always updated per segment (4 channels * [1 large window + 4 short windows]).

Bitrate Adjustment

It was previously said that the decisions for window switching and for coding mode selection are orthogonal in the sense that they do not depend on each other. Independent to these decisions is also the final step of the coding process that involves quantization, Huffman coding and bitstream composing; i.e. there is no feedback path. This fact has the advantage of reducing the whole coding delay to a minimum value (1024/48000 = 21.3 milliseconds) and also to avoid instabilities due to unorthodox coding situations.

The quantization process affects both spectral coefficients and scale factors. Spectral coefficients are clustered in bands, each band having the same step size or scale factor. Each step size is directly computed from the masking threshold corresponding to its band, as seen in 3.2, step 14. The quantized values, which

are integer numbers, are then converted to variable word length or Huffman codes. The total number of bits to code the segment, considering additional fields of the bitstream, is computed. Since the bitrate must be kept constant, the quantization process must be iteratively done till that number of bits is within predefined limits.

5 After the number of bits needed to code the whole segment, considering the basic masking threshold, the degree of adjustment is dictated by a buffer control unit. This control unit shares the deficit or credit of additional bits among several segments, according to the needs of each one.

The technique of the bitrate adjustment routine is represented by the flowchart of FIG. 9. It may be seen that after the total number of available bits to be used by the current segment is computed, an iterative procedure tries to find a factor α such that if all the initial thresholds are multiplied by this factor, the final total number of bits is smaller then and within an error δ of the available number of bits. Even if the approximation curve is so hostile that α is not found within the maximum number of iterations, one acceptable solution is always available.

The main steps of this routine are as follows. First, an interval including the solution is found. Then, a loop seeks to rapidly converge to the solution. At each iteration, the best solution is updated.

In order to use the same procedure for segments coded with large and short windows, in this latter case, the coefficients of the 4 short windows are clustered by concatenating homologue bands. Scale factors are clustered in the same.

The bitrate adjustment routine calls another routine that computes the total number of bits to represent all the Huffman coded words (coefficients and scale factors). This latter routine does a spectrum partitioning according to the amplitude distribution of the coefficients. The goal is to assign predefined Huffman code books to sections of the spectrum. Each section groups a variable number of bands and its coefficients are Huffman coded with a convenient book. The limits of the section and the reference of the code book must be sent to the decoder as side information.

25 The spectrum partitioning is done using a minimum cost strategy. The main steps are as follows. First, all possible sections are defined -the limit is one section per band- each one having the code book that best matches the amplitude distribution of the coefficients within that section. As the beginning and the end of the whole spectrum is known, if K is the number of sections, there are K-1 separators between sections. The price to eliminate each separator is computed. The separator that has a lower price is eliminated (initial prices may be negative). Prices are computed again before the next iteration. This process is repeated till a maximum allowable number of sections is obtained and the smallest price to eliminate another separator is higher than a predefined value.

Aspects of the processing accomplished by quantizer/rate-loop 206 in FIG. 2 will now be presented. In the prior art, rate-loop mechanisms have contained assumptions related to the monophonic case. With the shift from monophonic to stereophonic perceptual coders, the demands placed upon the rate-loop are increased.

The inputs to quantizer/rate-loop 206 in FIG. 2 comprise spectral coefficients (i.e., the MDCT coefficients) derived by analysis filter bank 202, and outputs of perceptual model 204, including calculated thresholds corresponding to the spectral coefficients.

Quantizer/rate-loop 206 quantizes the spectral information based, in part, on the calculated thresholds and the absolute thresholds of hearing and in doing so provides a bitstream to entropy coder 208. The bitstream includes signals divided into three parts: (1) a first part containing the standardized side information; (2) a second part containing the scaling factors for the 35 or 56 bands and additional side information used for so-called adaptive-window switching, when used (the length of this part can vary depending on information in the first part) and (3) a third part comprising the quantized spectral coefficients.

45 A "utilized scale factor", Δ , is iteratively derived by interpolating between a calculated scale factor and a scale factor derived from the absolute threshold of hearing at the frequency corresponding to the frequency of the respective spectral coefficient to be quantized until the quantized spectral coefficients can be encoded within permissible limits.

50 An illustrative embodiment of the present invention can be seen in FIG. W. As shown at W01 quantizer/rate-loop receives a spectral coefficient, C_n , and an energy threshold, E, corresponding to that spectral coefficient. A "threshold scale factor", Δ_0 is calculated by

$$\Delta_0 = \sqrt{12E}$$

55 An "absolute scale factor", Δ_A , is also calculated based upon the absolute threshold of hearing (i.e., the quietest sound that can be heard at the frequency corresponding to the scale factor). Advantageously, an interpolation constant, α , and interpolation bounds α_{high} and α_{low} are initialized to aid in the adjustment of the utilized scale factor.

$$\alpha_{high} = 1$$

$$\alpha_{low} = 0$$

$$\alpha = \alpha_{\text{high}}$$

Next, as shown in W05, the utilized scale factor is determined from:

$$\Delta = \Delta_0^{\alpha} \times \Delta_A^{(1 - \alpha)}$$

Next, as shown in W07, the utilized scale factor is itself quantized because the utilized scale factor as computed above is not discrete but is advantageously discrete when transmitted and used.

$$\Delta = Q^{-1}(Q(\Delta))$$

Next, as shown in W09, the spectral coefficient is quantized using the utilized scale factor to create a "quantized spectral coefficient" $Q(C_r, \Delta)$.

$$Q(C_r, \Delta) = \text{NINT}\left(\frac{C_r}{\Delta}\right)$$

where "NINT" is the nearest integer function. Because quantizer/rate loop 206 must transmit both the quantized spectral coefficient and the utilized scale factor, a cost, C, is calculated which is associated with how many bits it will take to transmit them both. As shown in FIG. W11,

$$C = \text{FOO}(Q(C_r, \Delta), Q(\Delta))$$

where FOO is a function which, depending on the specific embodiment, can be easily determined by persons having ordinary skill in the art of data communications. As shown in W13, the cost, C, is tested to determine whether it is in a permissible range PR. When the cost is within the permissible range, $Q(C_r, \Delta)$ and $Q(\Delta)$ are transmitted to entropy coder 208.

Advantageously, and depending on the relationship of the cost C to the permissible range PR the interpolation constant and bounds are adjusted until the utilized scale factor yields a quantized spectral coefficient which has a cost within the permissible range. Illustratively, as shown in FIG. W at W13, the interpolation bounds are manipulated to produce a binary search. Specifically,

when $C > \text{PR}$, $\alpha_{\text{high}} = \alpha$,

alternately,

when $C < \text{PR}$, $\alpha_{\text{low}} = \alpha$.

In either case, the interpolation constant is calculated by:

$$\alpha = \frac{\alpha_{\text{low}} + \alpha_{\text{high}}}{2}$$

The process then continues at W05 iteratively until the C comes within the permissible range PR.

STEREOPHONIC DECODER

The stereophonic decoder has a very simple structure. Its main functions are reading the incoming bit-stream, decoding all the data, inverse quantization and reconstruction of RIGHT and LEFT channels. The technique is represented in FIG. 12.

Illustrative embodiments may comprise digital signal processor (DSP) hardware, such as the AT&T DSP16 or DSP32C, and software performing the operations discussed below. Very large scale integration (VLSI) hardware embodiments of the present invention, as well as hybrid DSP/VLSI embodiments, may also be provided.

Claims

1. A method of coding a digital input signal to provide a coded digital output signal, the method comprising the steps of:

sampling the digital input signal to create a frame of 2N input signal samples;

analyzing the frame of signal samples with an odd-frequency fast Fourier transform to provide a frame of 2N Fourier coefficients; and

outputting a coded signal comprising samples $X(k)$, each sample $X(k)$ provided by multiplying the real part of a Fourier coefficient, $F(k)$, by $\cos\left[\frac{\pi(2k + 1)(1 + N)}{4N}\right]$.

FIG. 1
(PRIOR ART)

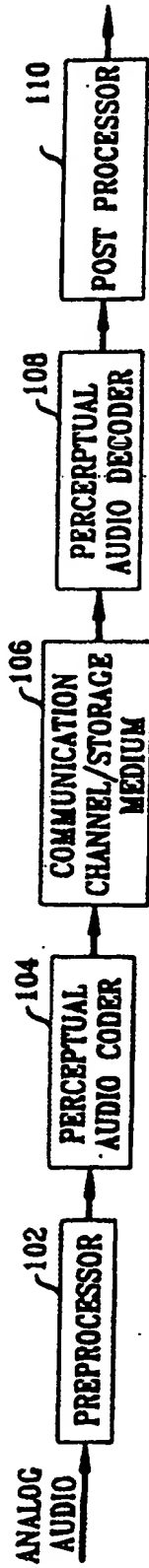


FIG. 2
(PRIOR ART)

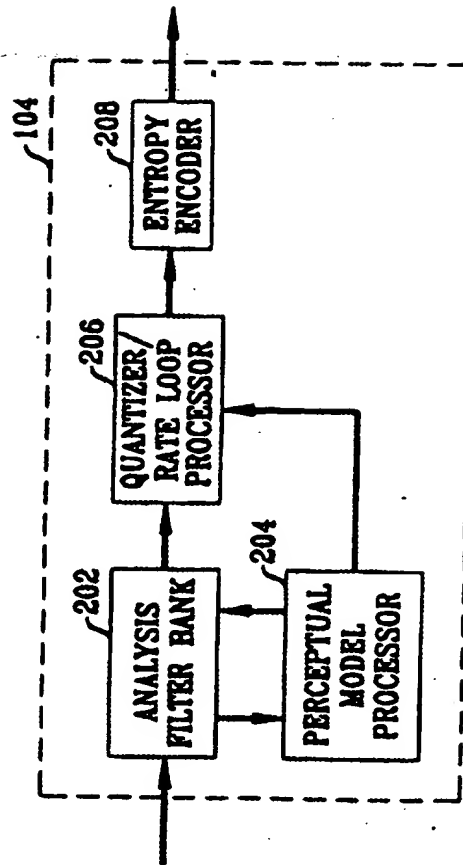


FIG. 3

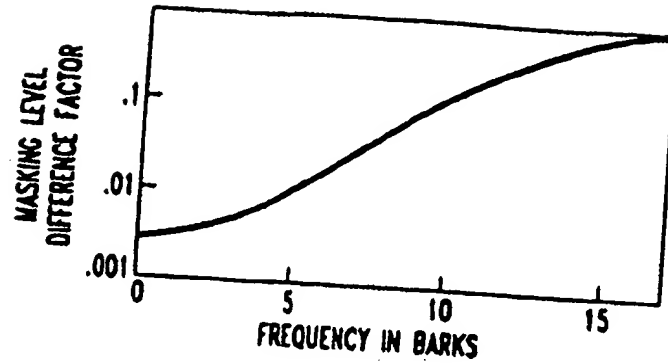


FIG. 8

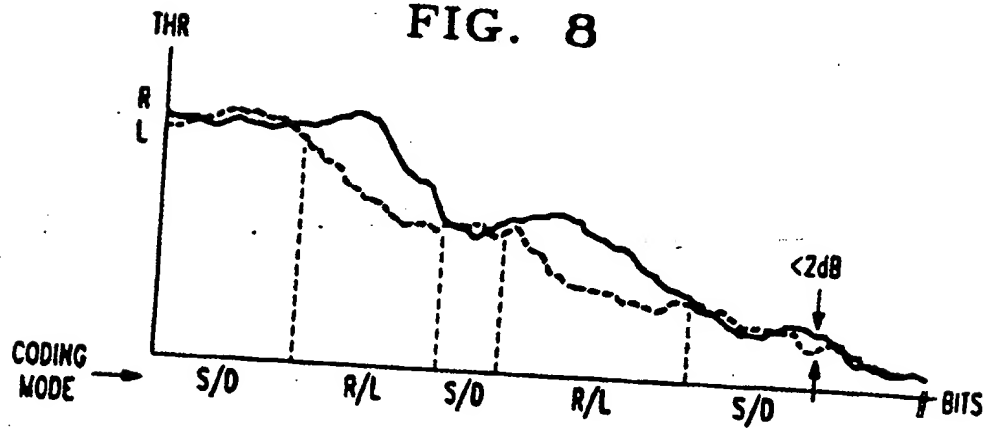


FIG. 10

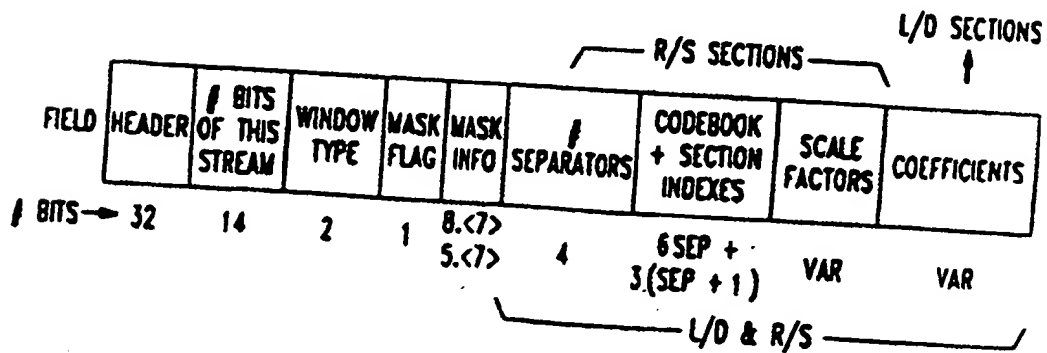


FIG. 4

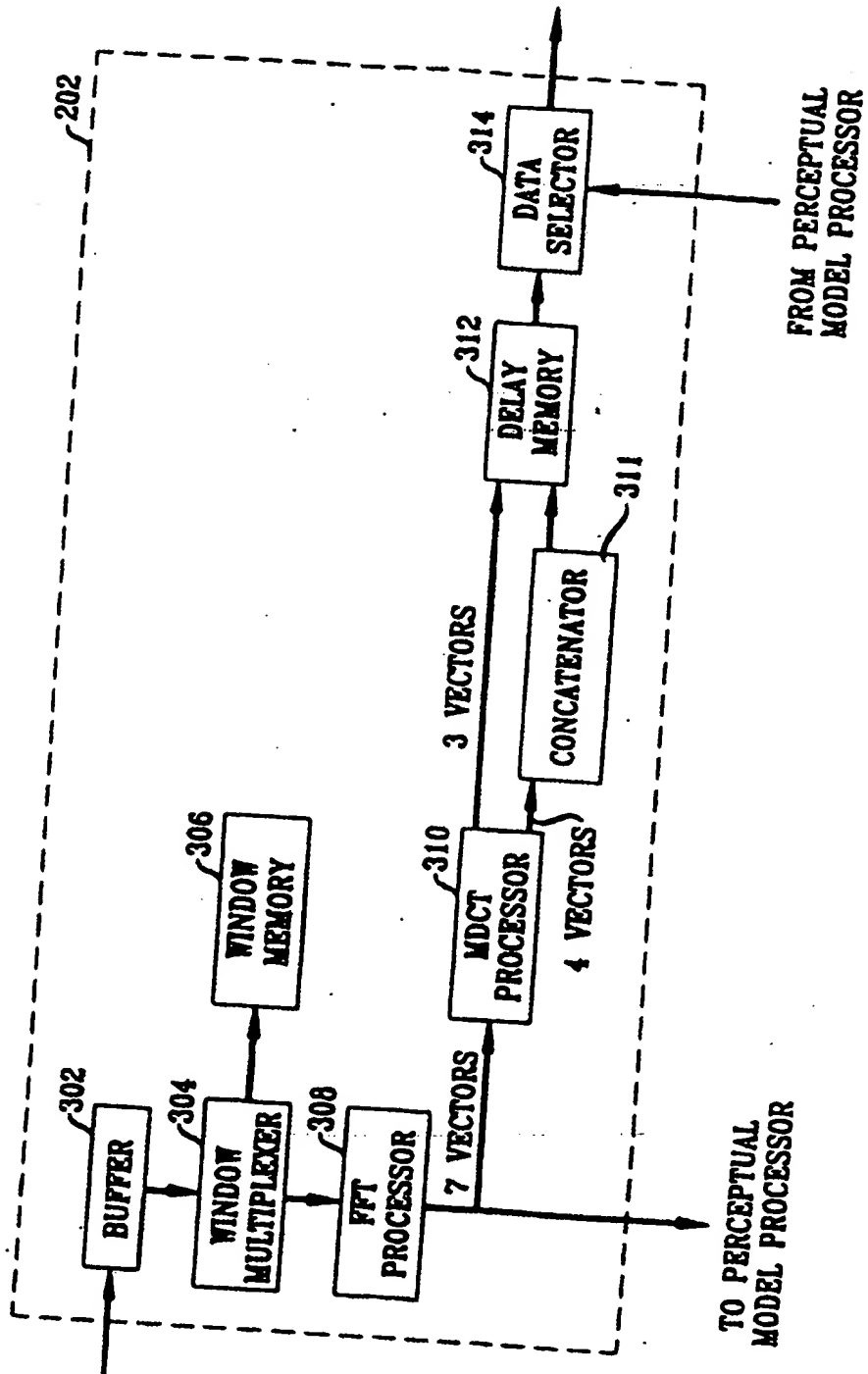


FIG. 5A

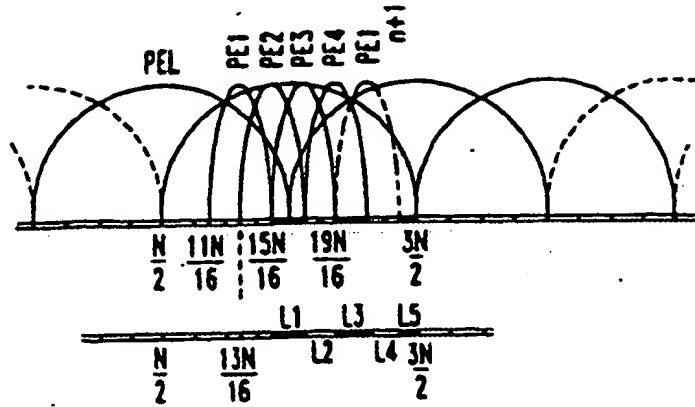


FIG. 5B

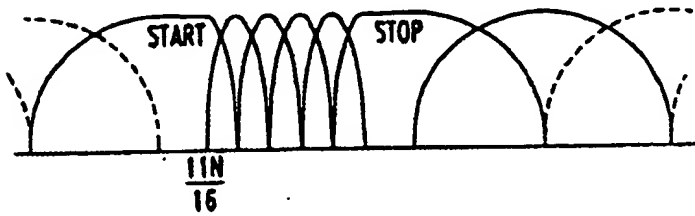


FIG. 5C

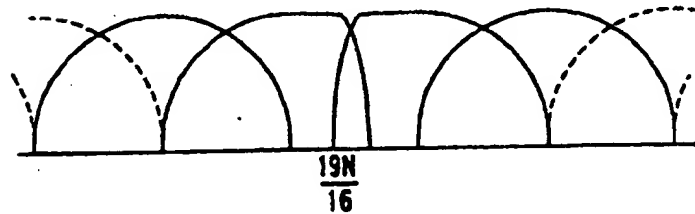


FIG. 5D

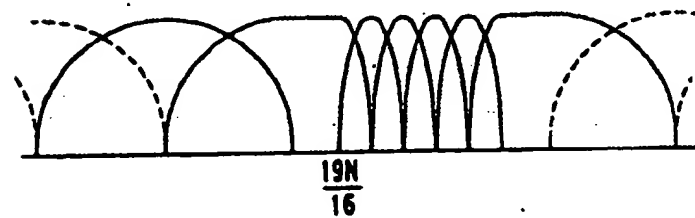


FIG. 5E

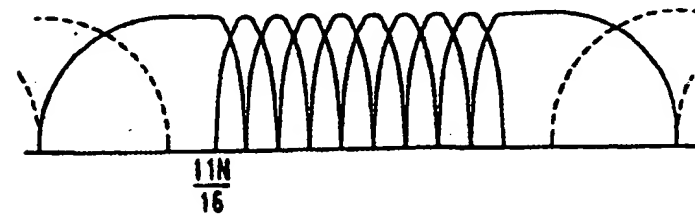


FIG. 6

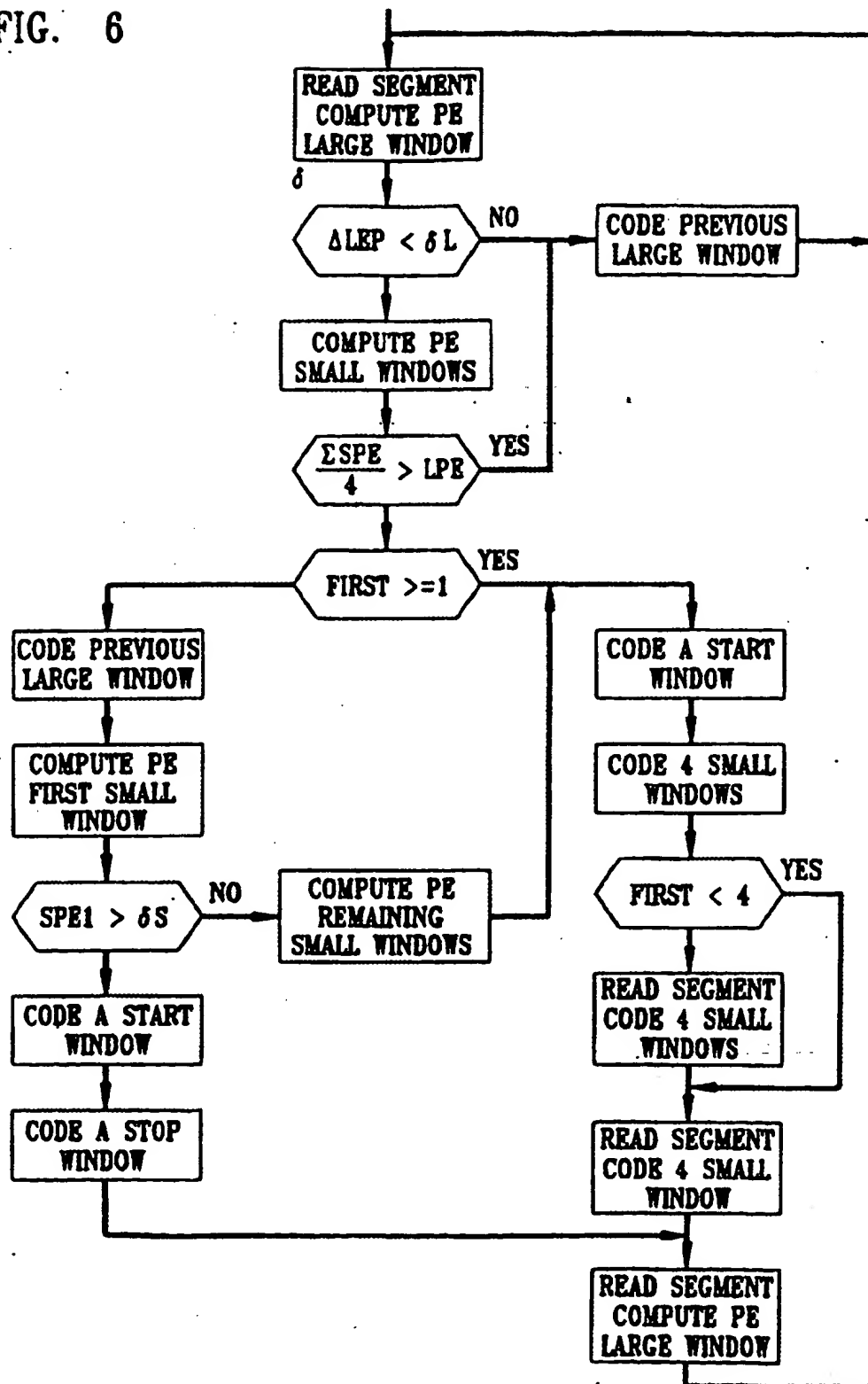


FIG. 7

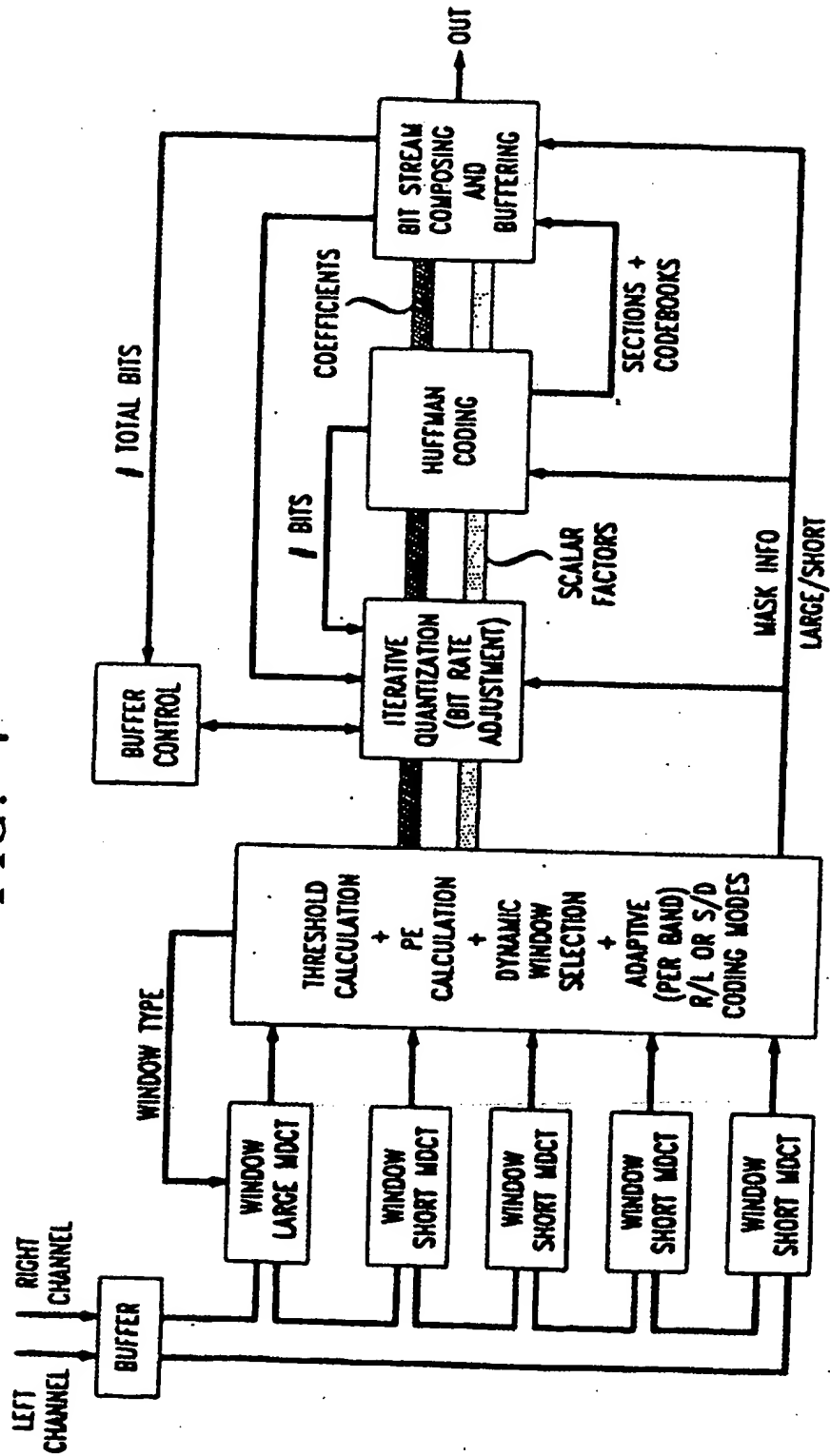


FIG. 9

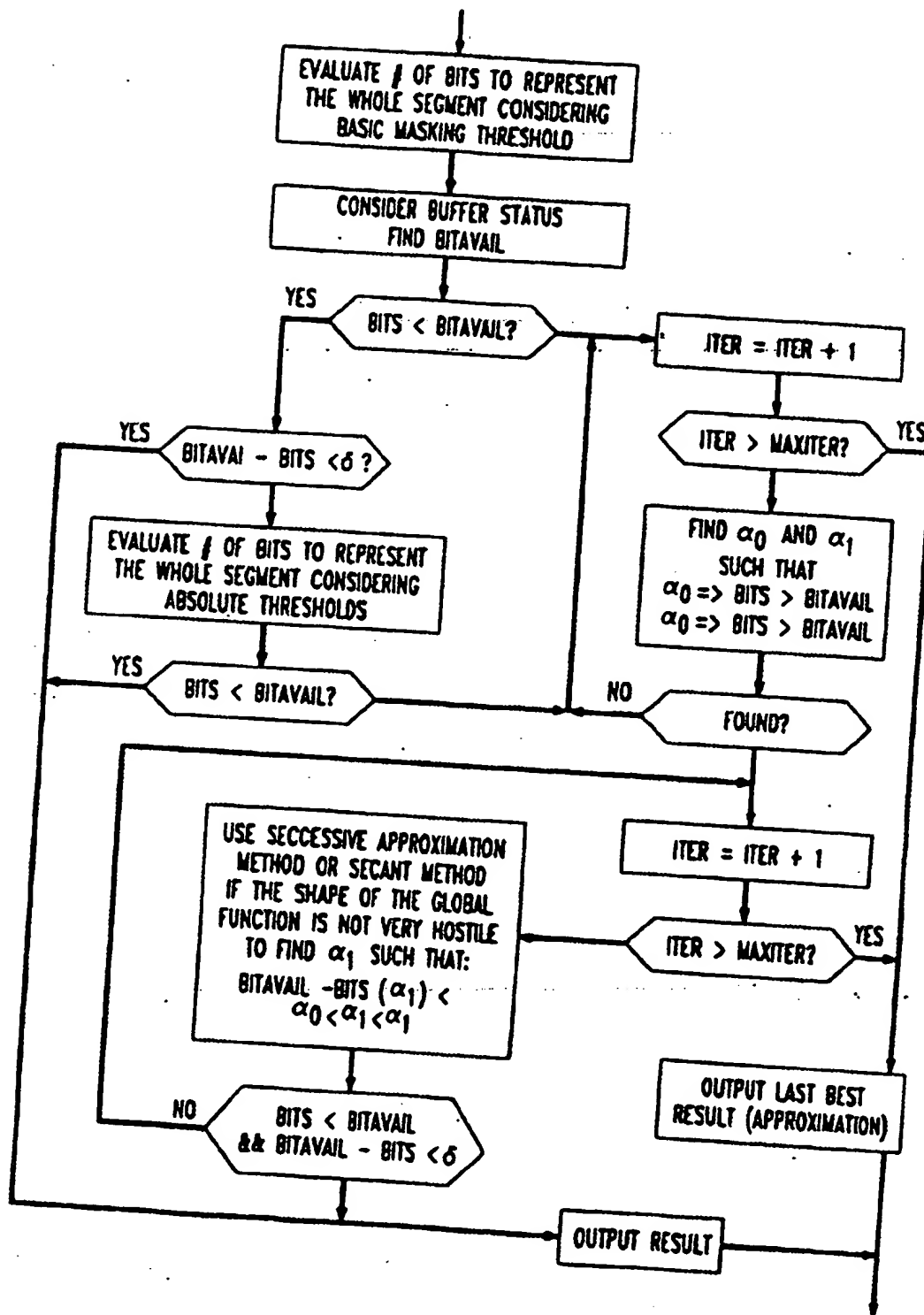
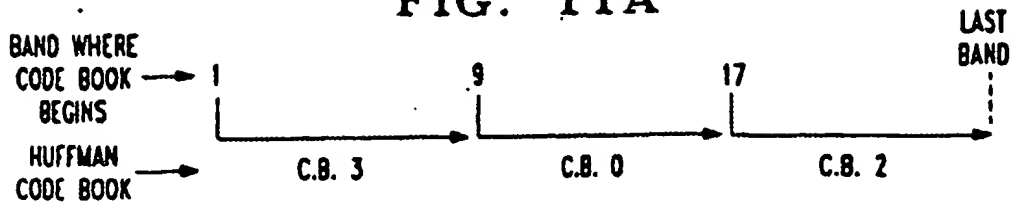


FIG. 11A

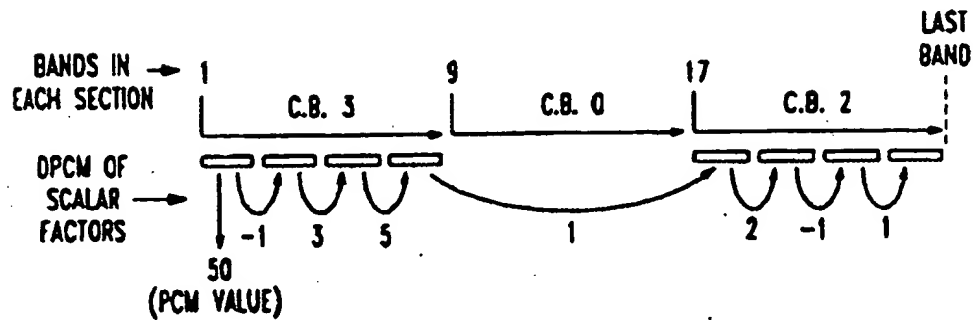


SEPARATORS R/S = K = 2

SECTION INDEX	CODE BOOK	BAND WHERE SECTION BEGINS
1	2	17
2	0	9
3	3	1

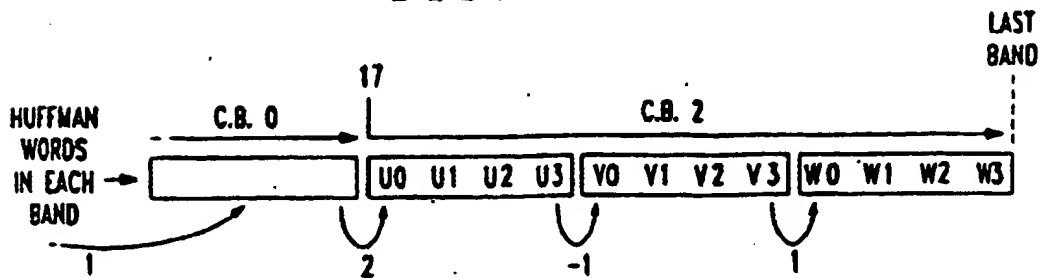
FIELD CB=INDX
IN THE STREAM:
[2, 17, 0, 9, 3]

FIG. 11B



FIELD SCALE FACTORS IN THE STREAM: [1, -1, 2, 1, 5, 3, -1, 50]

FIG. 11C



FIELD COEFFICIENTS IN THE STREAM: [W0-3, V0-3, U0-3, ...]

FIG. 12

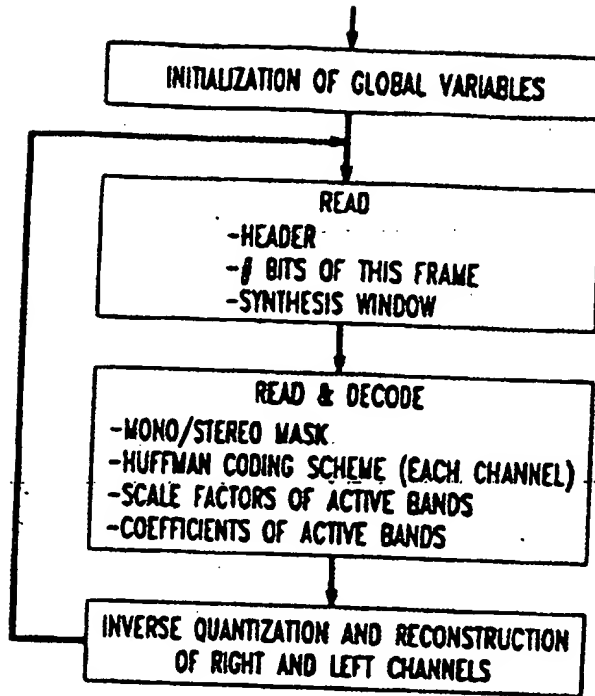


FIG. 13

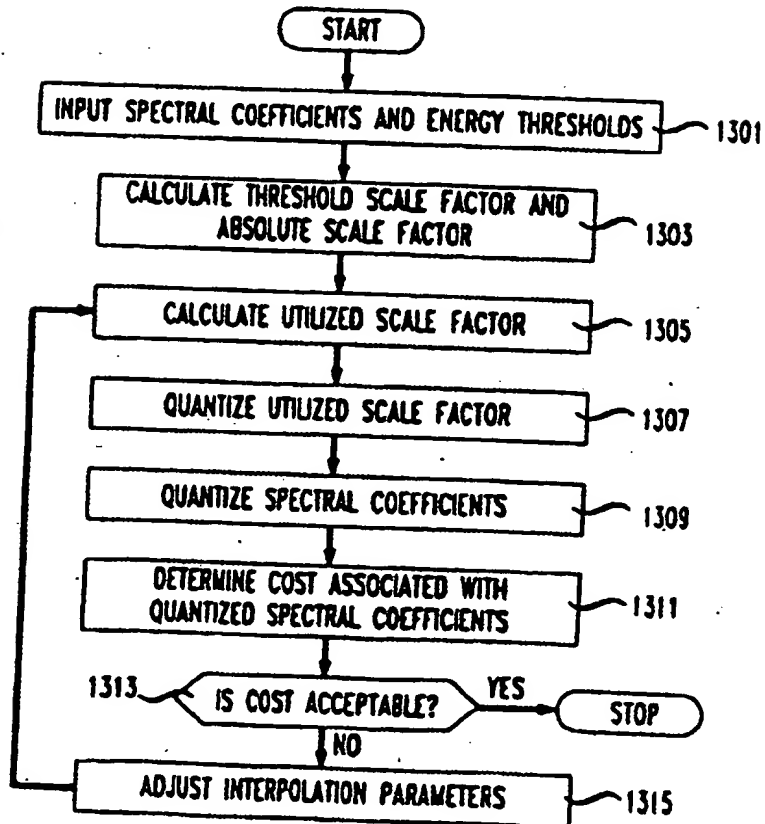
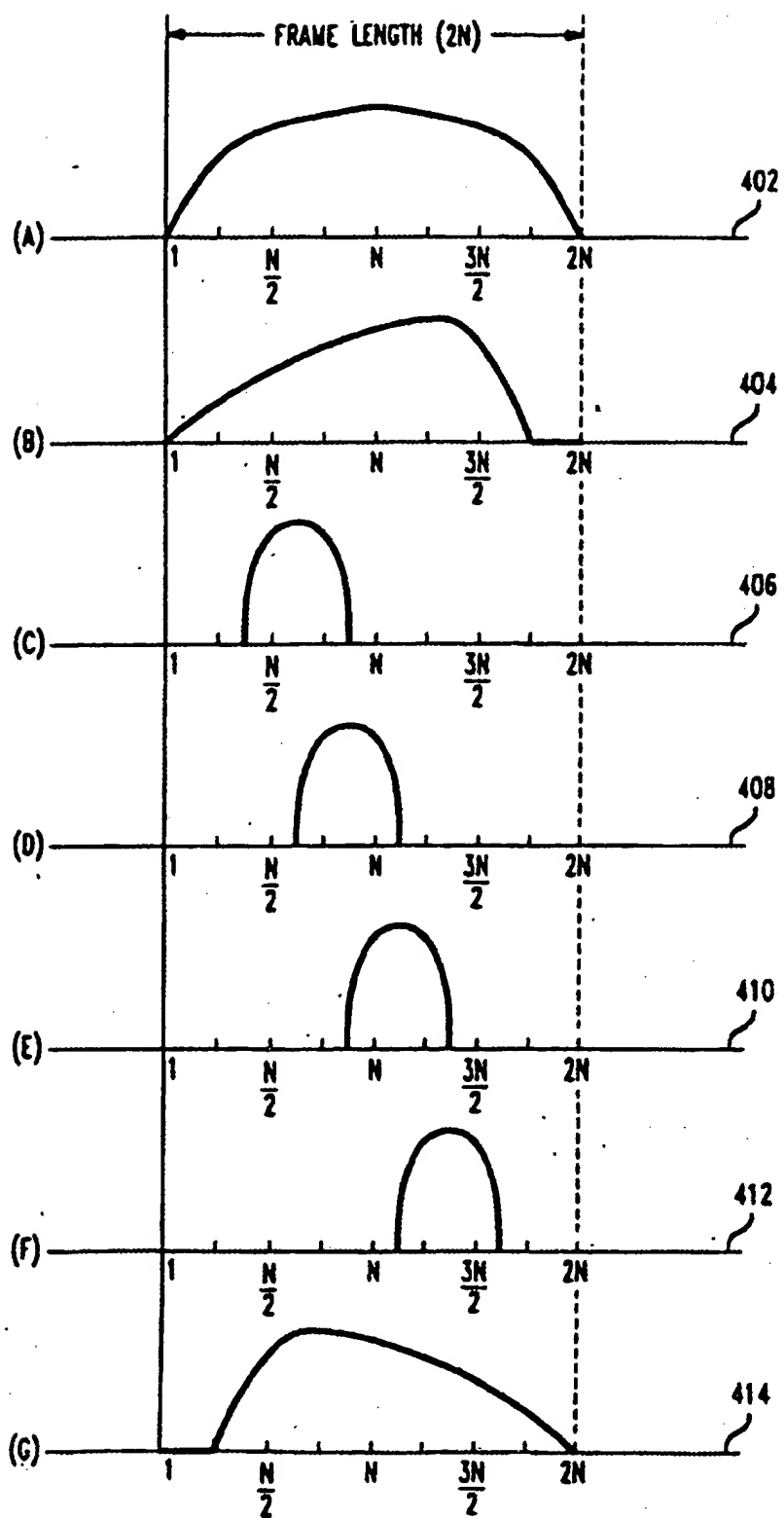


FIG. 14





European Patent
Office

EUROPEAN SEARCH REPORT

Application Number

DOCUMENTS CONSIDERED TO BE RELEVANT			EP 93301433.4
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
A	WO - A - 89/03 574 (FRAUNHOFER) * Abstract; claims *	1	H 04 B 1/10 H 03 M 7/02
A	WO - A - 89/09 985 (MIT) * Abstract; claims *	1	
A	EP - A - 0 193 143 (TELEFUNKEN) * Abstract; claims *	1	
A	FR - A - 2 484 755 (HENRY + DURET) * Abstract; claims *	1	
A	EP - A - 0 118 123 (TELEFUNKEN) * Abstract; claims *	1	
			TECHNICAL FIELDS SEARCHED (Int. Cl.5)
			G 10 L 1/00 G 10 L 5/00 G 10 L 7/00 H 03 M 7/00 H 04 B 1/00 H 04 J 3/00 H 04 Q 1/00
The present search report has been drawn up for all claims			
Place of search VIENNA	Date of completion of the search 04-06-1993	Examiner BLASL	
CATEGORY OF CITED DOCUMENTS		I : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date O : document cited in the application L : document cited for other reasons A : member of the same patent family, corresponding document	
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document			

EPO FORM 1503 (01.91) (P0401)



Europäisches Patentamt
European Patent Office
Office européen des brevets



Publication number:

0 506 111 A2

(12)

EUROPEAN PATENT APPLICATION

(21) Application number: 92105359.1

(51) Int. Cl.5: G06F 15/332

(22) Date of filing: 27.03.92

(30) Priority: 27.03.91 JP 63259/91

(43) Date of publication of application:
30.09.92 Bulletin 92/40

(84) Designated Contracting States:
DE FR NL

(71) Applicant: MITSUBISHI DENKI KABUSHIKI
KAISHA
2-3, Marunouchi 2-chome Chiyoda-ku
Tokyo(JP)

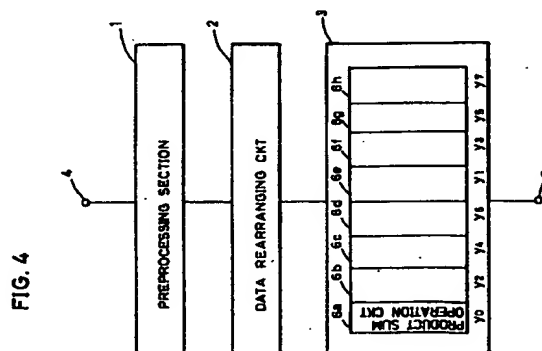
(72) Inventor: Uramoto, Shinichi, c/o Mitsubishi

Denki K.K. LSI
Kenkyusho, 1 Mizuhara 4-chome
Itami-shi, Hyogo-ken(JP)
Inventor: Inoue, Yoshitsugu, c/o Mitsubishi
Denki K.K. LSI
Kenkyusho, 1 Mizuhara 4-chome
Itami-shi, Hyogo-ken(JP)

(74) Representative: Prüfer, Lutz H.
PRÜFER & PARTNER, Patentanwälte,
Harthausen Strasse 25d
W-8000 München 90(DE)

(54) DCT/IDCT processor and data processing method.

(57) A one-dimensional discrete cosine transform (DCT) processor of N (N: positive integer)-term input data X includes a preprocessing section (1) for carrying out addition and subtraction of (i)th-term data $x(i)$ and (N - i)th-term data $x(N - 1)$ of input data X, and a unit (2, 3) for performing a product sum operation for sets of intermediate data subjected to preprocessing by addition and sets of intermediate data subjected to preprocessing by subtraction, respectively. The product sum operation unit includes a data rearranging unit (2) for outputting, in parallel and in order, bit data of the same figure of a set of data, a partial sum generator (41) for generating a partial sum by using the parallel bit data as an address, and an accumulator (42) for accumulating outputs of the partial sum generator. A one-dimensional inverse discrete cosine transform (IDCT) processor of N-term input data X includes a unit (2, 3) for performing a product sum operation of input data, and a postprocessing section (7) for carrying out addition and subtraction of 2-term data in a predetermined combination of an output of the product sum operation unit. The number of times of multiplication is reduced by utilizing inherent characteristics of coefficients of DCT/IDCT processing. Since the product sum operation is performed by a ROM table (43) and an adder (44), a faster multiplication is realized.



EP 0 506 111 A2

The present invention relates generally to data processors and data processing methods and, more particularly, to an apparatus and method for carrying out discrete cosine transform or inverse cosine transform of data.

In order to process video data at a high speed, high effective coding is carried out. In high effective coding, a data amount of a digital video signal is compressed with picture quality being maintained as high as possible. In high effective coding, a redundant component of the signal is first removed for efficient coding. For this purpose, orthogonal transform techniques are often employed. As one of the orthogonal transform techniques, discrete cosine transform DCT is provided. The DCT is implemented by a simple product sum operation using a cosine function as a coefficient. The DCT is defined by the following expression (1):

$$Y = AX \quad (1)$$

where X is an N-term column vector indicating input data, Y is an N-term column vector indicating output data, and A is N by N coefficient matrix represented by the following expression.

$$A(i, j) = \sqrt{\frac{2}{N}} \cdot C(i) \cdot \cos \frac{(2i+1)j}{2N} \pi$$

$$C(i) \begin{cases} = \sqrt{1/2} & (i = 0) \\ = 1 & (i \neq 0) \end{cases}$$

$$i, j = 0, 1, \dots, N-1$$

The expression (1) represents a case where input data X is of N terms. 2^m points are generally employed, where m is a natural number. A description will now be made on 8 point DCT where $N = 8$ ($m = 3$). As can be seen from the expression (1), DCT is a matrix operation, and in practice, this processing is realized by product sum operation.

Fig. 1 shows configuration of a conventional DCT processor. This DCT processor is described in, for example, IEEE, Proceedings of Custom Integrated Circuits Conference 89, 1989, pp. 24.4.1 to 24.4.4.

Referring to Fig. 1, the conventional DCT processor includes eight sum product operation units 100a to 100h arranged in parallel for calculating respective terms y_0 to y_7 of output data Y.

Each of product sum operation units 100a to 100h is of the same configuration and includes a parallel multiplier 101 for taking a product of input data x_i ($i = 0$ to 7) and a predetermined weighting coefficient, and an accumulator 102 for accumulating an output of parallel multiplier 101 to generate output data y_j ($j = 0$ to 7). Here, reference characters 101 and 102 generically denote respective components 101a to 101h and 102a to 102h. In the following description also, reference numerals having no suffixes generically denote corresponding elements.

Accumulator 102 includes a 2-input adder 103 for receiving an output of parallel multiplier 101 at its one input, and an accumulating register 104 for latching an output of adder 103. An output of register 104 is applied to an output terminal 106 and also to the other input of adder 103. Data y_j of the respective terms of output data Y are sequentially output through a selector not shown from output terminal 106. An operation will now be described.

Identical data are applied through an input terminal 105 to product sum operation units 100a to 100h. The following arithmetic operation is carried in each of product sum operation units 100a - 100h:

$$y_j = \sum_{i=0}^7 A(i, j) x_i$$

$$= \frac{1}{2} \sum_{i=0}^7 C(j) \cdot \left(\cos \frac{(2i+1)}{16} j\pi \right) \cdot x_i \quad \dots (2)$$

$$i, j = 0, 1, \dots, 7$$

For example, data y_0 of a zeroth term in an output data vector Y is calculated as follows in product sum operation unit 100a.

When receiving zeroth-term data x_0 (hereinafter referred to simply as input data) in an input data vector, parallel multiplier 101a outputs a product $A(0, 0) \cdot x_0$ of data x_0 and a coefficient $A(0, 0)$ to adder 103a. Register 104a is being reset, and the content thereof is 0. Accordingly, product $A(0, 0) \cdot x_0$ is output from adder 103a and then stored in register 104a.

When input data x_1 is applied, a product $A(1, 0) \cdot x_1$ is output from multiplier 101a. An output of adder 103a is $A(0, 0) \cdot x_0 + A(1, 0) \cdot x_1$ and stored in register 104a.

By repetition of such an operation, an output of accumulator 102a provided after application of input data x_7 is

$$\sum_i A(i, 0) \cdot x_i,$$

so that output data y_0 is obtained.

Similar calculation (which differs merely in values of a weighting coefficient $A(i, j)$) is carried out also in the remaining product sum operation units 100b - 100h, and output data $y_1 - y_7$ are obtained. These output data $y_0 - y_7$ are sequentially output through output terminal 106.

In contrast to the DCT operation, there is an inverse DCT operation for carrying out the inverse operation of the DCT operation. The inverse DCT (IDCT) is expressed as follows:

$$X = A'Y$$

where an input data vector X is obtained from an output data vector Y . That is, only the difference between the DCT operation and the IDCT operation is a difference between coefficients A and A' . Thus, in the configuration of Fig. 1, the IDCT operation can be carried out by changing the coefficients in parallel multipliers 101a - 101h.

In other words, the DCT and the IDCT can be carried out on the same hardware. An increase in hardware is only concerned with a control circuit (not shown) for making a selection between a coefficient for DCT and that for IDCT.

The above-described one-dimensional DCT operation can be expanded to a two-dimensional DCT operation. The two-dimensional DCT operation is obtained by making both input data vector X and output data vector Y be two-dimensional vectors.

Fig. 2 shows configuration of a conventional two-dimensional DCT (or IDCT) processor. Referring to Fig. 2, the processor includes a first one-dimensional DCT processing section 111a for subjecting input data from input terminal 105 to one-dimensional DCT processing, a transposition circuit 112 for rearranging rows and columns of an output of first one-dimensional DCT processing section 111a, and a second one-dimensional DCT processing section 111b for subjecting an output of transposition circuit 112 to one-dimensional DCT processing. First one-dimensional DCT processing section 111a performs a DCT (or IDCT) operation in a row direction, and second one-dimensional DCT processing section 111b performs a DCT (or IDCT) operation in a column direction.

Fig. 3 is a diagram showing configuration of the transposition circuit of Fig. 2. Referring to Fig. 3, transposition circuit 112 includes a buffer memory 121 and an address generation circuit 122 for generating write/read addresses of buffer memory 121. Buffer memory 121 receives output data of first-one-dimensional DCT processing section 111a through an input terminal 125 and sequentially stores the same therein in accordance with an address signal from address generation circuit 122. Also, buffer memory 121

applies corresponding data from an output terminal 126 to second one-dimensional DCT processing section 111b in accordance with an address signal from address generation circuit 122. An operation will now be described. Input data X and output data Y are two dimensional, the elements of which are each represented by $x(i, j)$ and $y(i, j)$, $i, j = 0, 1 \dots 7$.

Input data are applied in the order of rows to first one-dimensional DCT processing section 111a. More specifically, input data are applied to input terminal 105 in the order of 8-term row vectors $x(0, j)$, $x(1, j)$, ... $x(7, j)$.

First one-dimensional DCT processing section 111a performs the DCT operation for each row vector to output intermediate data Z. At that time, first DCT processing section 111a outputs intermediate data of row vectors in the order of rows, i.e., $z(0, j)$, $z(1, j)$ Accordingly, a DCT operation in the row direction of input data X is carried out.

As shown in Fig. 3, transposition circuit 112 first stores the intermediate data from first DCT processing section 111a into buffer memory 121 in the order of receiving of the intermediate data (the order of rows).

Then, intermediate data Z are read in the order of columns, i.e., the order of column vectors $z(i, 0)$, $z(i, 1) \dots$ from buffer memory 121.

Intermediate data Z read in the order of columns are applied to second DCT processing section 111b. Second DCT processing section 111b carries out on the intermediate data one-dimensional DCT processing. Accordingly, data subjected to one-dimensional DCT processing in the column direction are output from second one-dimensional DCT processing section 111b. Output data Y from second one-dimensional DCT processing section 111b are output in the order of columns from output terminal 106. As a result, two-dimensional DCT shown by the following equation (3) is performed.

$$Y_{uv} = \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 C(u) \cdot C(v) \cdot \cos \frac{(2i+1)u\pi}{16} \cdot \cos \frac{(2j+1)v\pi}{16} \cdot X_{ij} \quad \dots (3)$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & (u, v = 0) \\ 1 & (u, v \neq 0) \end{cases}$$

First and second DCT processing sections 111a and 111b carry out the same processing except for coefficients in the parallel multiplying circuits. If multiplication coefficients of first and second DCT processing sections 111a and 111b are changed, two-dimensional IDCT shown by the following equation (4) is carried out.

$$X_{ij} = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u) \cdot C(v) \cdot \cos \frac{(2i+1)u\pi}{16} \cdot \cos \frac{(2j+1)v\pi}{16} \cdot Y_{uv} \quad \dots (4)$$

The DCT processing and IDCT processing as shown above include a product sum operation. A product operation of this product sum operation is carried out by the parallel multipliers shown in Fig. 1. A multiplier in general requires a large number of adders and the like and has a large scale. Thus, there is a disadvantage that a conventional DCT processor requiring a plurality of parallel multipliers is not allowed to be sized-down.

In a semiconductor integrated circuit for carrying out a synchronization operation, the upper limit of operation speed is determined by a worst delay path (the path which provides a maximum delay). In the conventional configuration, the worst delay path is established by a parallel multiplier, and the operation speed depends on processing speed of the parallel multiplier. It is thus difficult to implement a fast DCT processing and a fast IDCT processing.

One object of the present invention is to provide a down-sized data processor which operates at a high speed.

Another object of the present invention is to provide a data processor for carrying out at least one of DCT and IDCT at a high speed.

A further object of the present invention is to provide a data processing method for carrying out at least one of DCT and IDCT at a high speed.

A data processor according to the present invention reduces the number of times of multiplication by utilizing characteristics inherent to a DCT operation or IDCT operation. A product sum operation is carried out by a successive operation employing a combination of a memory and an adder.

Since the number of times of multiplication is reduced and no parallel multipliers are employed, the DCT operation and IDCT operation are carried out at a high speed with fewer circuit components.

The foregoing and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

Fig. 1 is a diagram showing configuration of a conventional one-dimensional DCT processor.

Fig. 2 is a diagram showing configuration of a conventional two-dimensional DCT processor.

Fig. 3 is a diagram showing configuration of a transposition circuit of Fig. 2.

Fig. 4 is a diagram showing configuration of a one-dimensional DCT processor being one embodiment of the present invention.

Fig. 5 is a diagram showing an example of configuration of a preprocessing section shown in Fig. 4.

Fig. 6 is a diagram showing an example of modification of the preprocessing section of Fig. 5.

Fig. 7A is a diagram showing an example of configuration of a data rearranging circuit of Fig. 4.

Fig. 7B is a diagram showing the contents of a shift register of Fig. 7A.

Fig. 8 is a diagram showing an example of configuration of a product sum operation circuit of Fig. 4.

Fig. 9 is a diagram of an example of modification of the product sum operation circuit of Fig. 8.

Fig. 10 is a diagram showing an example of modification of the one-dimensional DCT processor of Fig. 4.

Fig. 11 is a diagram showing configuration of a one-dimensional IDCT processor being another embodiment of the present invention.

Fig. 12 is a diagram showing configuration of a one-dimensional DCT/IDCT processor being still another embodiment of the present invention.

Fig. 13 is a diagram showing configuration of a two-dimensional DCT processor being still another embodiment of the present invention.

Fig. 14 is a diagram showing configuration of a two-dimensional IDCT processor being still another embodiment of the present invention.

Fig. 15 is a diagram showing configuration of a two-dimensional DCT/IDCT processor being still another embodiment of the present invention.

Fig. 16 is a diagram showing configuration of a semiconductor integrated circuit apparatus including the DCT processor of the present invention.

Fig. 17 is a diagram showing an example of modification of the semiconductor integrated circuit of Fig. 16.

Fig. 4 schematically shows configuration of a one-dimensional DCT processor being one embodiment of the present invention.

Referring to Fig. 4, the processor includes a preprocessing section 1 for receiving input data x_i from an input terminal 4 to preprocess the received input data x_i on the basis of characteristics inherent to DCT operation, a data rearranging circuit 2 for rearranging data output from preprocessing section 1, and a product sum operation section 3 for carrying out a product sum operation on data from data rearranging circuit 2.

This processor carries out an eight-point DCT operation. Thus, product sum operation section 3 includes eight product sum operation circuits 6a - 6h. Respective product sum operation circuits 6a - 6h provide respective output data $y_0, y_2, y_4, y_6, y_1, y_3, y_5$ and y_7 to sequentially apply the output data to an output terminal 5, (the sequential application unit is not shown in the figure).

A description will now be made on the principle of an 8-point one-dimensional DCT processing

operation of the present invention before a detailed description of configuration of each section. If the relationship between input data x_i ($i = 0, 1, \dots, 7$) and output data y_j ($j = 0, 1, \dots, 7$) shown in equations (1) and (2) is expressed in a matrix form, the following representation (5) is obtained:

$$\begin{array}{c} 5 \\ 10 \\ 15 \\ 20 \end{array}
 \begin{array}{c} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \\ Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{array}
 =
 \begin{array}{c} A \quad A \quad A \quad A \quad A \quad A \quad A \quad A \\ B \quad C \quad -C \quad -B \quad -B \quad -C \quad C \quad B \\ A \quad -A \quad -A \quad A \quad A \quad -A \quad -A \quad A \\ C \quad -B \quad B \quad -C \quad -C \quad B \quad -B \quad C \\ D \quad E \quad F \quad G \quad -G \quad -F \quad -E \quad -D \\ E \quad -G \quad -D \quad -F \quad F \quad D \quad G \quad -E \\ F \quad -D \quad G \quad E \quad E \quad -G \quad D \quad -F \\ G \quad -F \quad E \quad -D \quad D \quad -E \quad F \quad -G \end{array}
 \begin{array}{c} x_0 \\ x_2 \\ x_4 \\ x_6 \\ x_1 \\ x_3 \\ x_5 \\ x_7 \end{array}
 \dots (5)$$

where

$$\begin{array}{c} 25 \\ 30 \end{array}
 \begin{array}{l} A = \frac{1}{2} \cos \frac{\pi}{4}, B = \frac{1}{2} \cos \frac{\pi}{8}, C = \frac{1}{2} \sin \frac{\pi}{8}, D = \frac{1}{2} \cos \frac{\pi}{16}, \\ E = \frac{1}{2} \cos \frac{3\pi}{16}, F = \frac{1}{2} \sin \frac{3\pi}{16}, G = \frac{1}{2} \sin \frac{\pi}{16} \end{array}$$

In derivation of the above relation (5), the well-known characteristics of trigonometric function such as $\cos \pi/4 = 1/\sqrt{2}$, $\cos (\pi \pm \theta) = -\cos \theta$, $\cos (\pi/2 \pm \theta) = \sin$ and the like are utilized. For example, $\cos (3\pi/8) = \sin (\pi/8)$ and the like are utilized.

In the above relation (5), a coefficient matrix is horizontally symmetrical with respect to columns. By use of this symmetry, relation (5) can be transformed to the following representation (6):

$$\begin{array}{c} 35 \\ 40 \\ 45 \\ 50 \end{array}
 \begin{array}{c} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \\ Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{array}
 =
 \begin{array}{c} A \quad A \quad A \quad A \\ B \quad C \quad -C \quad -B \\ A \quad -A \quad -A \quad A \\ C \quad -B \quad B \quad -C \\ D \quad E \quad F \quad G \\ E \quad -G \quad -D \quad -F \\ F \quad -D \quad G \quad E \\ G \quad -F \quad E \quad -D \end{array}
 \begin{array}{c} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \\ x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{array}
 \dots (6)$$

If a comparison is made between the above relations (5) and (6), it is apparent that the number of times of multiplication for acquiring output data y_j is reduced to a half in relation (6) as compared to relation (5). DCT processing in accordance with relation (6) is carried out in this embodiment.

With reference to Fig. 4, preprocessing section 1 generates the following eight intermediate data from input data x_i sequentially applied from input terminal 4 by selectively carrying out addition or subtraction.

The intermediate data are:

$(x_0 + x_7), (x_1 + x_6), (x_2 + x_5), (x_3 + x_4),$
 $(x_0 - x_7), (x_1 - x_6), (x_2 - x_5)$ and $(x_3 - x_4).$

5

The result of preprocessing from preprocessing section 1 is represented in finite word length. In the following description, it is assumed that the preprocessing result is indicated by 8-bit data in two's complement notation.

In order to calculate output data y_j by using the preprocessed data from preprocessing section 1, the matrix operation of relation (6) is carried out.

10 With respect to output data y_2 , for example, the following relation (7) is carried out:

$$\begin{aligned}
 y_2 &= B \cdot (x_0 + x_7) + C \cdot (x_1 + x_6) \\
 &\quad - C \cdot (x_2 + x_5) - B \cdot (x_3 + x_4) \\
 &= \sum_{k=1}^4 B_k \cdot z_k \quad \dots (7)
 \end{aligned}$$

20

where

$$\begin{aligned}
 25 \quad B_1 &= B, B_2 = C, B_3 = -C, B_4 = -B \\
 z_1 &= (x_0 + x_7), z_2 = (x_1 + x_6) \\
 z_3 &= (x_2 + x_5), z_4 = (x_3 + x_4)
 \end{aligned}$$

30 Data rearranging circuit 2 of Fig. 4 receives preprocessing results z_k ($k = 1, \dots, 4$) from preprocessing section 1. When receiving four necessary preprocessing results z_k , data rearranging circuit 2 outputs the least significant bits of respective four preprocessing results z_k in parallel to product sum operation section 3. The parallel bit output is carried out sequentially in bit figure until the most significant bit is output.

Product sum operation circuit 6a for output data y_2 carries out an operation in accordance with a relation (8) which is a further equivalent transformation of relation (7).

35

$$y_2 = \sum_{n=1}^7 \left[\sum_{k=1}^4 B_k \cdot z_{kn} \right] 2^{-n} + \sum_{k=1}^4 B_k \cdot (-z_{k0}) \quad \dots (8)$$

40

where z_{kn} is n th-bit data of preprocessing result z_k , and z_{k0} is the most significant bit of z_k . That is, $z_k <0|7> = (z_{k0}, z_{k1}, \dots, z_{k7})$. Data z_k is obtained in the following relation (9):

45

$$z_k = -z_{k0} + \sum_{n=1}^7 z_{kn} \cdot 2^{-n} \quad \dots (9)$$

50

It should be noted that data z_k is data of 8 bits represented in two's complement notation. Therefore, equations (7) and (8) are mathematically totally equivalent to each other except for a difference in order of product sum operations.

4-bit data z_{1n}, z_{2n}, z_{3n} and z_{4n} are applied in parallel to product sum operation circuit 6a for data y_2 . The values of coefficients B_1, B_2, B_3 and B_4 can be calculated in advance. Product sum operation circuit 6a stores therein a partial sum (10) shown below in the form of a ROM table and outputs a corresponding partial sum with 4-bit data z_{1n}, z_{2n}, z_{3n} and z_{4n} used as an address.

55

$$\sum_{k=1}^4 B_k \cdot z_{kn} \quad (n = 0, \dots, 7) \quad \dots (10)$$

This partial sum is accumulated by an internal accumulator, so that output data y2 is output from terminal 5. Although the sign of the most significant bit b_{n0} is negative, this sign can be converted to be positive for addition operation in two's complement notation.

Referring to Fig. 4, in product sum operation section 3, product sum operation circuits 6a - 6d apply the operation to the same data z_k in parallel to produce output data y0, y2, y4 and y6. Product sum operation circuits 6e - 6h apply the operation to the same data w_k in parallel to produce output data y1, y3, y5 and y7.

Output data y0 to y7 are sequentially output in this order from output terminal 5 by a selector not shown. A description will now be given on a detailed configuration of each section shown in Fig. 1.

Fig. 5 shows configuration of preprocessing section 1 shown in Fig. 4. Preprocessing section 1 includes an input circuit 21 for receiving input data xi sequentially applied from input terminal 4. Input circuit 21 outputs input data xp and xq in a predetermined combination under control by a control circuit 25. Here, a relation $p + q = 7$ is satisfied. Input circuit 21 can be formed of a tapped shift register. Data at a desired stage can be read by selecting a tap under control by control circuit 25 by employing, for example, a multiplexer.

Preprocessing section 1 further includes a 2-input adder 22 for adding outputs of input circuit 21, a subtractor 23 for subtracting outputs of input circuit 21, and an output circuit 24 for selecting one of respective outputs of adder 22 and subtractor 23 under control by control circuit 25. Adder 22 and subtractor 23 carry out addition and subtraction for the applied data under control by control circuit 25.

Output circuit 24 preferably alternately selects the output of adder 22 and that of subtractor 23. An operation will now be described.

Input circuit 21 receives input data X to sequentially output sets of data (x0, x7), (x1, x6), (x2, x5), and (x3, x4).

Adder 22 adds the data of each set. Adder 22 sequentially outputs data z_k , i.e., (x0 + x7), (x1 + x6), (x2 + x5) and (x3 + x4).

Subtractor 23 sequentially outputs data w_k , i.e., (x0 - x7), (x1 - x6), (x2 - x5) and (x3 - x4).

Output circuit 24 alternately outputs data z_k and data w_k .

Parallel multiplication circuits 6a to 6d of Fig. 1 carry out an operation in accordance with data z_k , while parallel multiplication circuits 6e to 6h carry out an operation in accordance with data w_k .

Output circuit 24 alternately outputs addition data z_k and subtraction data w_k . This makes it possible to produce output data y0 to y7 in this order from product sum operation section 3 and implement a pipelined architecture for processing data in synchronization with a clock signal.

In that case, it is unnecessary that adder 22 and subtractor 23 carry out an arithmetic operation simultaneously. Accordingly, as shown in Fig. 6, an arithmetic unit 26 for alternately performing the adding processing and the subtracting processing under control by control circuit 25 may be employed. Output circuit 24 does not have to have a selecting function in the configuration of Fig. 6. Output circuit 24 is required to have a function of buffering and latching (in the case of a clock synchronizing operation) an output of arithmetic unit 26. In the configuration of Fig. 6, since the addition and subtraction are carried out in a single arithmetic unit 26, the circuit scale is reduced.

Such configuration may be employed that intermediate data w_k is output after all intermediate data z_k are output from preprocessing section 1.

Fig. 7A shows configuration of data rearranging circuit 2 of Fig. 1. Data rearranging circuit 2 includes an input circuit 31 for receiving intermediate data from a terminal 500, a shift register 32 for sequentially storing therein data from input circuit 31, and a selector 33 for sequentially reading four intermediate data stored in shift register 32 from the least significant bit.

After alternately receiving intermediate data z_k and intermediate data w_k and outputting all of intermediate data z_k in advance, input circuit 31 sequentially outputs intermediate data w_k . This configuration can easily be implemented by using a register for storing intermediate data w_k therein.

When intermediate data w_k are applied after all of intermediate data z_k are applied, input circuit 31 sequentially outputs intermediate data from terminal 500. Input circuit 31, however, has a function of latching intermediate data w_k until the reading of intermediate data z_k by the selector is completed. A data acceptance, latching and output operation of input circuit 31 is controlled by a control circuit 34.

Shift register 32 stores therein four intermediate data from input circuit 31. Shift register 32 includes

four 8-bit registers 32a-32d in a row direction as shown in Fig. 7B. Fig. 7B shows a state where four intermediate data z_1 to z_4 are stored in shift register 32.

Intermediate data z_k from input circuit 31 are sequentially stored in registers 32a - 32d. After all intermediate data z_1 to z_4 are stored in register 32, data of registers 32a - 32d are read in parallel sequentially from the respective least significant bits. Such configuration can be implemented by shift registers capable of shifting in both row and column directions. Even by use of a shift register capable of shifting only in the row direction, if a register stage is selected by selector 33, the data rearranging operation can be realized.

A data bit shifting operation of shift register 32 is controlled by control circuit 34. Selector 33 reads data of 4 bits in parallel from shift register 32 under control by control circuit 34.

Four-bit data z_{kn} are output from a terminal 501 in the configuration of Fig. 7A.

Fig. 8 shows configuration of product sum operation circuit 6. Referring to Fig. 8, product sum operation circuit 6 includes a partial sum generating circuit 41 for generating a partial sum in accordance with data from terminal 501, and an accumulator 42 for accumulating an output of partial sum generating circuit 41.

Partial sum generating circuit 41 includes an ROM (Read Only Memory) 43 for receiving 4-bit data z_{kn} and as an address signal. ROM 43 stores the partial sum shown in, for example, equation (10) in the form of table and, when supplied with 4-bit data z_{kn} , ROM 43 outputs a corresponding value. By constructing this partial sum generating circuit 41 in the form of the ROM table, a partial sum can be generated at a high speed without any multiplication.

Accumulator 42 includes an adder 44 for receiving a partial sum from partial sum generating circuit 41 at its one input, an accumulating register 45 for storing an output of adder 44, and a shifter 46 for shifting an output of register 45 by predetermined bits to apply the shifted output to the other input of adder 44. Output data y_j is applied from shifter 46 to terminal 5. A description will now be made on an operation thereof, taking output data y_2 as an example.

Four-bit data z_{kn} are applied in turn from the least significant bit to product sum generating circuit 41. Product sum generating circuit 41 sequentially outputs a partial sum

$$\sum_{k=1}^4 B_k \cdot z_{kn}$$

from ROM 43.

First, a partial sum

$$\sum_{k=1}^4 B_k \cdot z_{k1}$$

is stored in register 45.

Then, a partial sum

$$\sum_{k=1}^4 B_k \cdot z_{k1}$$

is output from partial sum generating circuit 41.

Shifter 46 shifts the contents of register 45 by one bit. Accordingly, an output of shifter 46 is shown as below:

$$[\sum_{k=1}^4 B_k \cdot z_{k7}] 2^{-1}$$

The output of adder 44 is shown as below:

$$\sum_{k=1}^4 B_k \cdot z_{k6} + \left(\sum_{k=1}^4 B_k \cdot z_{k7} \right) \cdot 2^{-1}$$

By sequentially repeating this operation, the following output (11) is stored in register 45.

$$\sum_{n=1}^7 \left[\sum_{k=1}^4 B_k \cdot z_{kn} \right] \cdot 2^{-n+1} \dots (11)$$

If data Z_{k0} is applied, the contents of register 45 is the value shown in relation (8) since the data represented by the above relation (11) is shifted by one bit and then added by adder 44. After that, the shifting operation by shifter 46 is stopped and the contents of register 45 is read, whereby output data y_2 is obtained.

The operation of partial sum generating circuit 41 and accumulator 42 is carried out by control circuit 47.

Fig. 9 shows another configuration of a product sum operation circuit. The product sum operation circuit shown in Fig. 9 is different from the configuration shown in Fig. 8 in that a partial sum generating circuit 41 includes two ROMs 43a and 43b and an adder 48 for adding outputs of ROMs 43a and 43b.

ROM 43a receives higher order bits; while ROM 43b receives lower order bits. In this configuration, partial sums P and Q shown in the following equations are made by ROMs 43a and 43b.

$$P = \sum_{k=1}^2 B_k \cdot z_{kn}$$

$$Q = \sum_{k=3}^4 B_k \cdot z_{kn}$$

In the configuration of Fig. 9, the number of words to be stored into the ROMs is drastically reduced. This is because the number of words to be stored is determined by the number of bits of an address signal and increased in proportion to two's power of the bit number.

In the above configuration, product sum operation section 3 includes eight product sum operation circuits 6a - 6h. Intermediate data z_k and w_k are not calculated simultaneously. When intermediate data w_k are calculated after all intermediate data z_k are calculated and then output data y_0, y_2, y_4 and y_6 are calculated, product sum operation section 3 can be formed of four product sum operation circuits 6a - 6d as shown in Fig. 10.

Product sum operation circuits 6a - 6d calculate y_0 and y_1, y_2 and y_3, y_4 and y_5 , and y_6 and 7, respectively. The contents of an ROM for partial sum generation is changed in accordance with intermediate data z_k and w_k . If the ROM is structured in bank architecture, the change of the coefficient table can easily be realized.

A description will now be made on a structure for an IDCT operation with reference to Fig. 11. Referring to Fig. 11, an 8-point one-dimensional IDCT processor includes a data rearranging circuit 2 for rearranging data from an input terminal 4, a product sum operation section 3 for performing a product sum operation in accordance with an output of data rearranging circuit 2, and a postprocessing section 7 for carrying out addition and subtraction of a predetermined combination of outputs of product sum operation section 3.

Data rearranging circuit 2 and postprocessing section 7 are of the same configurations as those of rearranging circuit 2 and preprocessing section 1 of Fig. 4, respectively. An operation will now be described.

Input data y_j ($j = 0, 1, \dots, 7$) applied to terminal 4 undergoes an IDCT processing, so that output data x_i ($i = 0, 1, \dots, 7$) is transmitted to terminal 5. The relationship between data y_j and x_i is represented in the following matrix form (12).

$$\begin{matrix}
 & \begin{matrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{matrix} & = & \begin{bmatrix} A & B & A & C & D & E & F & G \\ A & C & -A & -B & E & -G & -D & -F \\ A & -C & -A & B & F & -D & G & E \\ A & -B & A & -C & G & -F & E & -D \\ A & -B & A & -C & -G & F & -E & D \\ A & -C & -A & B & -F & D & -G & -E \\ A & C & -A & -B & -E & G & D & F \\ A & B & A & C & -D & -E & -F & -G \end{bmatrix} & \begin{matrix} y_0 \\ y_2 \\ y_4 \\ y_6 \\ y_1 \\ y_3 \\ y_5 \\ y_7 \end{matrix} & \dots & (12)
 \end{matrix}$$

where

$$\begin{aligned}
 A &= \frac{1}{2} \cos \frac{\pi}{4}, B = \frac{1}{2} \cos \frac{\pi}{8}, C = \frac{1}{2} \sin \frac{\pi}{8}, D = \frac{1}{2} \cos \frac{\pi}{16}, \\
 E &= \frac{1}{2} \cos \frac{3\pi}{16}, F = \frac{1}{2} \sin \frac{3\pi}{16}, G = \frac{1}{2} \sin \frac{\pi}{16}
 \end{aligned}$$

This coefficient matrix is a transposed matrix of the coefficient matrix of equation (5). If the symmetry with respect to rows of the coefficient matrix of expression (12) is utilized, expression (12) is changed to the following equivalent expression (13).

$$\begin{matrix}
 & \begin{matrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{matrix} & = & \begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix} & \begin{matrix} y_0 \\ y_2 \\ y_4 \\ y_6 \end{matrix} & + & \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} & \begin{matrix} y_1 \\ y_3 \\ y_5 \\ y_7 \end{matrix} & \dots & (13)
 \end{matrix}$$

$$\begin{matrix}
 & \begin{matrix} x_7 \\ x_6 \\ x_5 \\ x_4 \end{matrix} & = & \begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix} & \begin{matrix} y_0 \\ y_2 \\ y_4 \\ y_6 \end{matrix} & - & \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} & \begin{matrix} y_1 \\ y_3 \\ y_5 \\ y_7 \end{matrix}
 \end{matrix}$$

It should be noted that there are only two types of the coefficient matrix of expression (13). Assume that these two types are M and N. The processor shown in Fig. 11 carries out an IDCT operation in accordance with expression (13).

Data rearranging circuit 2 receives data y_j ($j = 0, 1, \dots, 7$) from terminal 4 to rearrange data y_0, y_2, y_4 and y_6 and sequentially output the rearranged data from the least significant bit. That is, 4-bit data y_{0n}, y_{2n}, y_{4n} and y_{6n} ($n = 0, 1, \dots, 7$) of input data y_j are output from data rearranging circuit 2, so that generation and accumulation of partial sums are carried out. As intermediate data, the following data is output:

$$\sum_{n=1}^7 \left[\sum_{k=1}^4 B_k' \cdot Y_{k,n} \right] \cdot 2^{-n} + \sum_{k'=1}^4 B_{k'}' \cdot (-Y_{k',0})$$

5

This result corresponds to, for example, an intermediate term $M_2 = (A \cdot y_0 - C \cdot y_2 - A \cdot y_4 + B \cdot y_6)$ for x_2 . Then, data y_{1n}, y_{3n}, y_{5n} and y_{7n} ($n = 0, 1, \dots, 7$) are output from data rearranging circuit 2. The bit data is subjected to a product sum operation in product sum operation section 3. Accordingly, the remaining terms are obtained. For example, an intermediate term $N_2 = (F \cdot y_1 - D \cdot y_3 + G \cdot y_5 + E \cdot y_7)$ for x_2 is obtained. Intermediate terms M_i ($i = 0, 1, \dots, 7$) and N_i ($i = 0, 1, \dots, 7$) are output in turn from product sum operation section 3. From expression (13), the following relations are satisfied: $M_i = M_{7-i}$, and $N_i = N_{7-i}$.

10

Data rearranging circuit 2 may alternately output data bits (y_0, y_2, y_4, y_6) and data bits (y_1, y_3, y_5, y_7). Each of product sum operation circuits 6a - 6d calculates M_i ($= M_{7-i}$), and each of product sum operation circuits 6e-6h calculates N_i ($= N_{7-i}$).

15

Postprocessing section 7 obtains a sum of and a difference between intermediate data M_i and N_i to generate output data x_i and output the same to terminal 5. Accordingly, the following relation is obtained:

$$x_i = M_i + N_i \quad (i = 0, 1, 2, 3)$$

20

$$x_i = M_i - N_i \quad (i = 4, 5, 6, 7)$$

Postprocessing section 7 has the same configuration as that of Fig. 5 or 6. In that case, input circuit 21 sequentially or alternately receives intermediate terms M_i ($i = 0$ to 3), N_i ($i = 0$ to 3) to apply a desired combination of the terms to adder/subtractors 22, 23 (or 26). The order in which data are selected in the input circuit is made by control circuit 25. In this case, data may be applied in the order of $x_0, x_7, x_1, x_6, x_2, x_5, x_3, x_4$ to output circuit 24, and output circuit 24 may output the data in the order of x_0, x_1, \dots, x_7 .

25

This one-dimensional IDCT processor can also be structured such that a single product sum operation circuit 6 calculates both intermediate terms M_i and N_i ($i = 0$ to 3).

30

The product sum operation in DCT processing and that in IDCT processing are identical to each other except for their coefficient matrixes. Accordingly, as shown in Fig. 12, a processor capable of selectively performing the DCT processing and the IDCT processing is obtained.

Referring to Fig. 12, the processor includes a preprocessing section 1, a data rearranging circuit 2, a product sum operation section 3, a postprocessing section 7 and a control circuit 8.

35

Preprocessing section 1 has its input connected to an input terminal 4 and its output connected to data rearranging circuit 2. Data rearranging circuit 2 has its output connected to product sum operation section 3. Product sum operation section 3 has its output connected to an input of postprocessing section 7. The output of postprocessing section 7 is supplied through an output terminal 5. Product sum operation section 3 includes first to eighth product sum operation circuits 6a - 6h.

40

Control circuit 8 switches DCT operation and IDCT operation and also controls the operation of the respective circuits.

A description will now be made on an operation of the processor shown in Fig. 12. In the case of DCT processing, data is allowed to go intactly through postprocessing section 7. This causes the processor of Fig. 12 to function equally to the DCT processor shown in Fig. 4. That is, data input from input terminal 4 undergoes addition/subtraction in preprocessing section 1 and then rearranged in data rearranging circuit 2. The rearranged data is then transmitted in turn from lower order bits to the product sum operation section. The data subjected to a product sum operation shown in, for example, expression (5) in the product sum operation section passes through postprocessing section 7 and is then directly output from output terminal

50

5. In the case of inverse DCT processing, data passes intactly through preprocessing section 1, whereby the processor functions equally to the inverse DCT processor shown in Fig. 11 as follows. That is, the data input from input terminal 4 passes intactly through preprocessing section 1 and then rearranged in data rearranging circuit 2. The rearranged data is transmitted in turn from lower order bits to the product sum operation section. The data subjected to the product sum operation in the product sum operation section is transmitted to postprocessing section 7 and then subjected to addition/subtraction for calculating output data. The added/subtracted data is output from output terminal 5.

55

Changes in coefficients in product sum operation section 3 are made by control circuit 1. This is easily

realized by switching of banks of ROM or the like.

The above-described processor performs a one-dimensional DCT or IDCT operation. This processor can be developed to be able to perform a two-dimensional DCT or IDCT operation.

Fig. 13 shows configuration of a two-dimensional DCT processor according to the present invention. Referring to Fig. 13, the two-dimensional DCT processor includes a first one-dimensional DCT processing section 11a, a second one-dimensional DCT processing section 11b and a transposition circuit 12.

First, one-dimensional DCT processing section 11a carries out DCT processing with respect to rows, while second one-dimensional DCT processing section 11b carries out DCT processing with respect to columns. Transposition circuit 12 outputs in the order of columns the data applied in the order of rows. First and second processing sections 11a and 11b have the same configuration as that of the one-dimensional DCT processor shown in Fig. 1 and include a preprocessing section 1 (1a, 1b), a data rearranging circuit 2 (2a, 2b) and a product sum operation section 3 (3a, 3b). A description will now be made on a two-dimensional DCT processing of 8 x 8 points taken as an example.

If expression (3) is rewritten, the following expression (14) is obtained:

$$Y_{uv} = \sum_{j=0}^7 A(j, v) \cdot \left(\sum_{i=0}^7 A(i, u) \cdot x_{ij} \right) \dots \quad (14)$$

Input terminal 4 is supplied with input data in the order of rows. That is, 8-term row vector data $x(0, j)$, $x(1, j) \dots x(7, j)$ ($j = 0, 1, \dots, 7$) are applied in turn.

Preprocessing section 1a carries out preprocessing for the respective row vector data. For a zeroth row, for example, data $(x00 \pm x07)$, $(x01 \pm x06)$, $(x02 \pm x05)$ and $(x03 \pm x04)$ are output from preprocessing section 1a. Data rearranging circuit 2a rearranges four words (four addition data or four subtraction data) to output the rearranged data to product sum operation section 3a. Product sum operation section 3a applies a product sum operation to the applied data. The processing operation of data rearranging circuit 2a and product sum operation section 3a is the same as those of the processor shown in Fig. 4.

Accordingly, first one-dimensional DCT processing section 11a outputs in the order of rows 8-term row vector data R_k subjected to one-dimensional DCT processing with respect to a row direction. R_k is an 8-term row vector of $R_k = (R_{k0}, R_{k1}, \dots, R_{k7})$, where $k = 0, 1, \dots, 7$.

This intermediate data R_k is applied to transposition circuit 12 and stored therein in the order of rows. When 8-row intermediate data $R_0 - R_7$ are stored in transposition circuit 12, transposition circuit 12 outputs intermediate data in the order of columns to second one-dimensional DCT processing circuit 11b. The intermediate data stored in transposition circuit 12 is data which is subjected to an operation processing with respect to "i" in expression (14). Transposition circuit 12 outputs intermediate data in the order of columns. In the zeroth column, for example, data $R_{00}, R_{10}, R_{20}, \dots, R_{70}$ are read in turn.

Second one-dimensional DCT processing section 11b carries out the same preprocessing, the same data rearranging processing and the same product sum operation processing for each column as those of first one-dimensional DCT processing section 11a. Accordingly, second one-dimensional DCT processing section 11b outputs data subjected to addition with respect to "j" in expression (14). That is, 8-term column vector data are output in the order of columns from output terminal 5. The data appearing on output terminal 5 are data subjected to one-dimensional DCT processing in both row and column directions, i.e., two-dimensional DCT processing.

Like the transposition in the circuit shown in Fig. 2, transformation from rows to columns in transposition circuit 12 is realized by changing an address of a buffer memory in the row direction in data writing and in the column direction in data reading.

Also, the two-dimensional IDCT processing can be realized by expanding the one-dimensional IDCT processor shown in Fig. 11. Fig. 14 shows configuration of a two-dimensional IDCT processor.

Referring to Fig. 14, the two-dimensional IDCT processor includes a first one-dimensional IDCT processor 13a and a second one-dimensional IDCT processor 13b.

First, one-dimensional IDCT processor 13a includes a data rearranging circuit 2a, a product sum operation section 3a and a postprocessing section 7a. Second one-dimensional IDCT processor 13b includes a data rearranging circuit 2b, a product sum operation section 3b and a postprocessing section 7b. Both first and second one-dimensional IDCT processors 13a and 13b carry out the same processing as that of the one-dimensional IDCT processor shown in Fig. 11.

Input terminal 4 is supplied with input data in the order of rows. First IDCT processor 13a carries out an

IDCT processing with respect to rows.

Transposition circuit 12 sequentially stores therein intermediate data applied in the order of rows from first IDCT processor 13a and outputs the stored intermediate data in the order of columns.

Second IDCT processor 13b carries out an IDCT processing for the respective columns. Accordingly, output terminal 5 is supplied with the data subjected to the IDCT processing in both row and column directions, i.e., two-dimensional IDCT processing, in the order of columns.

Fig. 15 shows configuration of a two-dimensional DCT/IDCT processor being still another embodiment of the present invention. The processor of Fig. 15 includes a first one-dimensional DCT/IDCT processor 14a, a second one-dimensional DCT/IDCT processor 14b, and a transposition circuit 12 provided between processors 14a and 14b.

First and second processors 14a and 14b are of the same configuration as that of the processor shown in Fig. 12 and include a preprocessing section 1 (1a, 1b), a data rearranging circuit 2 (2a, 2b), a product sum operation section 3 (3a, 3b) and a postprocessing section 7 (7a, 7b).

In the configuration of Fig. 15, like the configuration shown in Fig. 12, if preprocessing sections 1a and 1b and postprocessing sections 7a and 7b are selectively set in a through state and coefficients (used in the partial sum generation circuit) of product sum operation sections 3a and 3b are selected, then two-dimensional DCT and IDCT processings can selectively be carried out.

An operation of the processor of Fig. 15 is identical to those of the processors of Figs. 13 and 14. One processing mode of the DCT processing and the IDCT processing is set by a control circuit not shown (corresponding to control circuit 8 of Fig. 12).

Although the foregoing description has not been concerned with implementation forms of the DCT processors, the use of the above-described configuration makes it possible to easily incorporate all of DCT (inverse DCT) functions integrally on a semiconductor integrated circuit.

It is also possible to incorporate all of the above-described DCT/inverse DCT functions integrally on a semiconductor integrated circuit and simultaneously incorporate functional circuitry having functions other than the DCT/inverse DCT functions integrally on one semiconductor substrate. Fig. 16 shows an example of use of a DCT processor which is incorporated integrally on one semiconductor substrate simultaneously with other functional circuitry.

Referring to Fig. 16, a semiconductor integrated circuit (semiconductor chip) 50 includes a DCT processor 51 and functional circuits 52, 53 and 54.

DCT processor 51 has such configuration as shown in Fig. 13 or 17. Functional circuits 52, 53 and 54 have different functions A, B and C, respectively. In application to video data processing, functions A, B and C include such functions necessary for image compression as quantization, variable length coding (entropy coding) and the like. The functions necessary for image compression are standardized by, for example, JPEG (Joint of Photographic Expert Group).

In the configuration shown in Fig. 16, DCT processor 51 is used in cooperation with (or in clock synchronization with) functional circuits 52, 53 and 54.

In the embodiment shown in Fig. 16, the functional circuits integrated together with the DCT processor are dedicated circuits having specific functions. The functional circuits are not limited to such dedicated circuits and may be integrated together with a microprocessor or a programmable functional block 56 such as a DSP (Digital Signal Processor) as shown in, for example, Fig. 17. Further, the DCT processor may be integrated together with a dedicated functional circuit 55 and programmable functional block 56 in combination as shown in Fig. 17.

The summary of principal technical effects of the present invention is as follows:

(i) Since the required number of times of multiplication is reduced by preprocessing in DCT processing or by postprocessing in IDCT processing, load on a product sum operation circuit is reduced.

(ii) Since a product sum operation is carried out by a memory and an adder, the scale of circuitry is substantively reduced.

(iii) Because of the above item (ii), a parallel multiplication circuit is unnecessary. Accordingly, when the entire processor performs a synchronizing operation, a higher operation speed on a worst delay path is easily achieved, facilitating a faster processing.

(iv) Since the effect of the above item (iii) facilitates an upgrading of a DCT (or IDCT) processor, this effect is greatly advantageous particularly in implementation of the present DCT (or IDCT) processor on a semiconductor integrated circuit, together with the effect of reducing the circuit scale.

Although the present invention has been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the present invention being limited only by the terms of the appended claims.

Claims

1. A processor having at least a function of carrying out one-dimensional discrete cosine transform (DCT) of N-term input data X, wherein said N is a positive integer, said processor comprising:
 - 5 preprocessing means (1; 1a, 1b) for carrying out addition and subtraction for each of sets of predetermined two terms of said input data X to generate a first set of addition data and a second set of subtraction data; and
 - matrix product means (2, 3) for obtaining a first matrix product of said first set of data from said preprocessing means and a predetermined first coefficient matrix, and a second matrix product of said
 - 10 second set of data and a predetermined second coefficient matrix, wherein an output of said matrix product means provides N-term output data subjected to DCT processing.
2. The processor of claim 1, wherein
 - said preprocessing means includes:
 - 15 set generating means (21) for generating a set of pth term data X(p) and qth term data X(q) of said input data X, where $p + q = N-1$, $0 \leq p < q \leq N-1$, and p and q are an integer;
 - addition means (22; 26) for carrying out addition of 2-term data output from said set generating means; and
 - subtraction means (23, 26) for carrying out subtraction of the 2-term data output from said set
 - 20 generating means.
3. The processor of claim 1 or 2, wherein
 - said matrix product means (2, 3) includes
 - storage means (32) for sequentially receiving said first set of data from said preprocessing means
 - 25 (1) to store the received data therein, each of said first set of data having a plurality of bits, and
 - parallel reading means (33) for reading, in parallel and in order, one-bit data in the same bit figure of all of said first set of data stored in said storage means.
4. The processor of one of claims 1 to 3, wherein
 - said matrix product means (2, 3) includes
 - storage means (32) for sequentially receiving said second set of data from said preprocessing means (1) to store the received data therein, each of said second set of data having a plurality of bits, and
 - 30 parallel reading means (33) for reading in parallel and in order, one-bit data in the same bit figure of all of said second set of data stored in said storage means.
5. The processor of claim 3 or 4, wherein
 - said matrix product means (2, 3) further includes
 - a plurality of product sum operation means (6a - 6d), and wherein each said product sum operation means includes
 - 40 table memory means (43) for receiving parallel bit data from said parallel reading means (33) as an address signal to output a corresponding partial sum, said table memory means (43) storing in advance a product sum of a corresponding coefficient and parallel bit data in a table form, and
 - accumulation means (42) for accumulating outputs of said table memory means (43), said
 - 45 accumulation means output providing a first set of output data of said N-term output data.
6. The processor of claim 5, wherein
 - said accumulation means (42) includes
 - addition means (44) for receiving an output of said table memory means at its one input,
 - 50 register means (45) for temporarily storing an output of said addition means therein, and
 - shift means (46) for shifting storage data in said register means by a bit to apply the shifted data to the other input of said addition means.
7. The processor of one of claims 1 to 6, wherein
 - said matrix product means (2, 3) further includes
 - 55 a plurality of product sum operation means (6e - 6h; 6a - 6d) for generating a second set of said N-term output data, and wherein
 - each said product sum operation means includes

table memory means (43) for receiving parallel bit data from said parallel reading means (33) as an address signal to output a corresponding partial sum, said table memory means (43) storing in advance a product sum of a corresponding coefficient and parallel bit data in a table form, and

accumulation means (42) for accumulating outputs of said table memory means to generate a second set of data of said N-term output data.

8. The processor of one of claims 5 to 7, wherein

said accumulation means (42) includes

addition means (44) for receiving an output of said table memory means at its one input,

register means (45) for temporarily storing an output of said addition means, and

shift means (46) for shifting storage data in said register means by a bit to apply the shifted data to the other input of said addition means, a final output of said shift means providing said second set of data of said N-term output data.

9. The processor of one of claims 1 to 8, further comprising:

a postprocessing section (7) for receiving an output of said matrix product means to carry out addition and subtraction of predetermined 2-term data of the received N-term data and generate first and second sets of output data; and

control means (8) for enabling one of said preprocessing section (1) and said postprocessing section.

10. The processor of claim 9, wherein

said postprocessing section (7) includes means (22, 23; 26) for carrying out addition and subtraction of (2i)th-term data $Y(2i)$ and (2i + 1)th-term data $Y(2i + 1)$ of N-term output data Y of said matrix product means (2, 3) wherein said i is an integer of $0 \leq i \leq N/2 - 1$.

11. The processor of claim 10, wherein

the addition of said data $Y(2i)$ and $Y(2i + 1)$ indicates (i)th-term output data $Z(i)$, and the subtraction of said data $Y(2i)$ and $Y(2i + 1)$ indicates (N - i - 1)th-term output data $Z(N - i - 1)$.

12. The processor of one of claims 1 to 11, further comprising:

transposition means (12) for sequentially receiving output data of said matrix product means (2, 3) to store the received data therein, transpose a matrix formed by the stored data and sequentially output N-term intermediate data;

second preprocessing means (1b) having the same configuration as that of said preprocessing means, for receiving an output of said transposition means to carry out addition and subtraction for each of predetermined 2-term sets of said N-term intermediate data; and

second matrix product means (2b, 3b) having the same configuration as that of said matrix product means, for performing a product operation of output data of said second preprocessing means and a predetermined second coefficient matrix, an output of said second matrix product means indicating data subjected to two-dimensional DCT processing.

13. The processor of one of claims 9 to 12, further comprising:

second postprocessing means (7b) having the same configuration as that of said postprocessing means, for receiving an output of said second matrix product means, and

second control means (8) for enabling one of said second preprocessing means (1b) and said second postprocessing means.

14. The processor of one of claims 1 to 13, wherein

said processor is incorporated integrally in an integrated circuit (50) with functional circuitry (52, 53, 54; 55, 56).

15. The processor of one of claims 1 to 14, wherein

said N is 8.

16. A processor having at least a function of carrying out one-dimensional inverse discrete cosine transform (DCT) of N-term input data Y, wherein said N is a positive integer, said processor comprising:

matrix product means (2, 3) for dividing said N-term input data Y into a first set of input data and a

second set of input data and carrying out a product operation of said first set of input data and a first coefficient matrix and a product operation of said second set of input data and a second coefficient matrix, to generate a first set of intermediate data M_i and a second set of intermediate data N_i , wherein said i is an integer of $0 \leq i \leq N/2 - 1$; and

postprocessing means (7; 7a, 7b) for carrying out addition and subtraction of two intermediate data in a predetermined relationship in said first set of intermediate data and said second set of intermediate data from said matrix product means to generate first and second sets of output data X_i .

17. The processor of claim 16, wherein

said postprocessing means (7; 7a, 7b) includes means (22, 23; 26) for carrying out addition and subtraction of said first set of (i)th-term intermediate data M_i and said second set of (i)th-term intermediate data N_i ; and

addition data ($M_i + N_i$) indicates (i)th-term data of N-term output data, and subtraction data ($M_i - N_i$) indicates (N - i - 1)th-term data of said N-term output data.

18. The processor of claim 16 or 17, wherein

each said intermediate data is represented by a plurality of bits, and

said matrix product means (2, 3) includes

storage means (32) for dividing said N-term input data Y into a first set of input data Y (2i) and a second set of input data Y (2i + 1) to store each set of the input data therein,

first reading means (33) for reading in parallel one-bit data in the same order of said first set of all input data Y (2i) from said storage means,

second reading means (33) for reading in parallel one-bit data in the same bit figure of said second set of all input data Y (2i + 1) from said storage means,

first product sum operation means (6a - 6d) for carrying out a product sum operation of parallel bit data from said first reading means and a corresponding coefficient of said first coefficient matrix, to generate said first set of output data, and

second product sum operation means (6e - 6h) for carrying out a product sum operation of parallel bit data from said second reading means and a corresponding coefficient of said second coefficient matrix, to generate said second set of output data.

19. The processor of claim 18, wherein

said first and second product sum operation means include a plurality of operation circuits each related to one term of said output data, each said operation circuit means (6a - 6h) including

table memory means (43) for receiving parallel bit data as an address signal to output the result of the product sum operation with the corresponding coefficient, said table memory means storing in advance data indicating the result of the product sum operation in a table form, and

accumulation means (42) for accumulating outputs of said table memory means.

20. The processor of claim 19, wherein

said accumulation means includes

2-input addition means (44) for receiving an output of said table memory means (43) at its one input,

register means (45) for temporarily storing an output of said addition means, and

shift means (46) for shifting storage data in said register means by a bit to apply the shifted data to the other input of said addition means, a final output of said shift means indicating output data of an associated term.

21. The processor of one of claims 16 to 20, further comprising:

preprocessing means (1; 1a, 1b) for carrying out addition and subtraction of a predetermined set of 2-term data Y (j), Y (N - j - 1) of said N-term input data Y to generate a first set of addition data and a second set of subtraction data, said first set of said addition data and said second set of said subtraction data being applied as said first and second sets of input data to said matrix product means; and

control means (8) for enabling one of said preprocessing means and said postprocessing means.

22. The processor of one of claims 16 to 21, further comprising: transposition means (12) for sequentially receiving N-term N output data from said postprocessing means (7; 7a, 7b) to store the received data

therein, then transpose the stored data and output the transposed data;

second matrix product means (2b, 3b) of the same configuration as that of said matrix product means, for receiving an output of said transposition means; and

second postprocessing means (7b) of the same configuration as that of said postprocessing means, for receiving an output of said matrix product means,

an output of said second postprocessing means indicating data subjected to two-dimensional IDCT processing of N by N points.

23. The processor of one of claims 16 to 22, wherein said N is 8.

24. The processor of one of claims 16 to 23, wherein said processor (51) is incorporated integrally in an integrated circuit (50) so as to operate in cooperation with other functional circuitry (52, 53, 54; 55, 56).

25. A method of processing one dimensional discrete cosine transform of N points, X , wherein said N is 2^m , said m being a natural number, said method comprising the steps of:

carrying out addition and subtraction of each 2-term data in predetermined relationship in said input data X , to generate a first set of addition data and a second set of subtraction data, said first and second sets including $N/2$ -term data;

carrying out a product operation of said first set of data and a first coefficient matrix to generate a first set of output data;

carrying out a product operation of said second set of data and a second coefficient matrix to generate a second set of output data; and

outputting said first set of output data and said second set of output data in a predetermined order.

26. The method of claim 25, wherein

said step of generating said first and second sets of output data includes the step of generating a corresponding partial sum by reference to a table memory, using an applied data as an address signal.

27. The method of claim 25 or 26, wherein

said 2-term data in said predetermined relationship are (i) th-term data $x(i)$ and $(N - i - 1)$ th-term data $x(N - i - 1)$, wherein said i is an integer of $0 \leq i \leq N/2 - 1$.

28. A method of carrying out one-dimensional inverse discrete cosine transform of N points, wherein said N is 2^m , said m being a natural number, said method comprising the steps of:

receiving N -term input data Y to generate a first set of input data of even-term data $Y(2i)$ and a second set of input data of odd-term input data $Y(2i + 1)$, wherein said i is an integer of $0 \leq i \leq N/2 - 1$;

carrying out a product operation of said first set of input data and a first coefficient matrix to generate a first set of intermediate data $M(i)$;

carrying out a product operation of said second set of input data and a second coefficient matrix to generate a second set of intermediate data $N(i)$;

carrying out addition and subtraction of said first set of intermediate data $M(i)$ and said second set of intermediate data $N(i)$ to generate a first set of addition data and a second set of subtraction data; and

outputting said first set of said addition data and said second set of said subtraction data in a predetermined order.

29. The method of claim 28, wherein

said addition data is a sum of data $M(i)$ and data $N(i)$, said addition data of $M(i) + N(i)$ indicating i th-term data $X(i)$ of N -term output data; and

said subtraction data is a difference between data $M(i)$ and data $N(i)$, said subtraction data of $M(i) - N(i)$ providing $(N - i - 1)$ th-term output data $X(N - i - 1)$.

30. The method of claim 28 or 29, wherein

said step of generating said intermediate data $M(i)$ and $N(i)$ includes the step of generating a corresponding partial sum by reference to a table memory, using the applied data as an address.

FIG.1

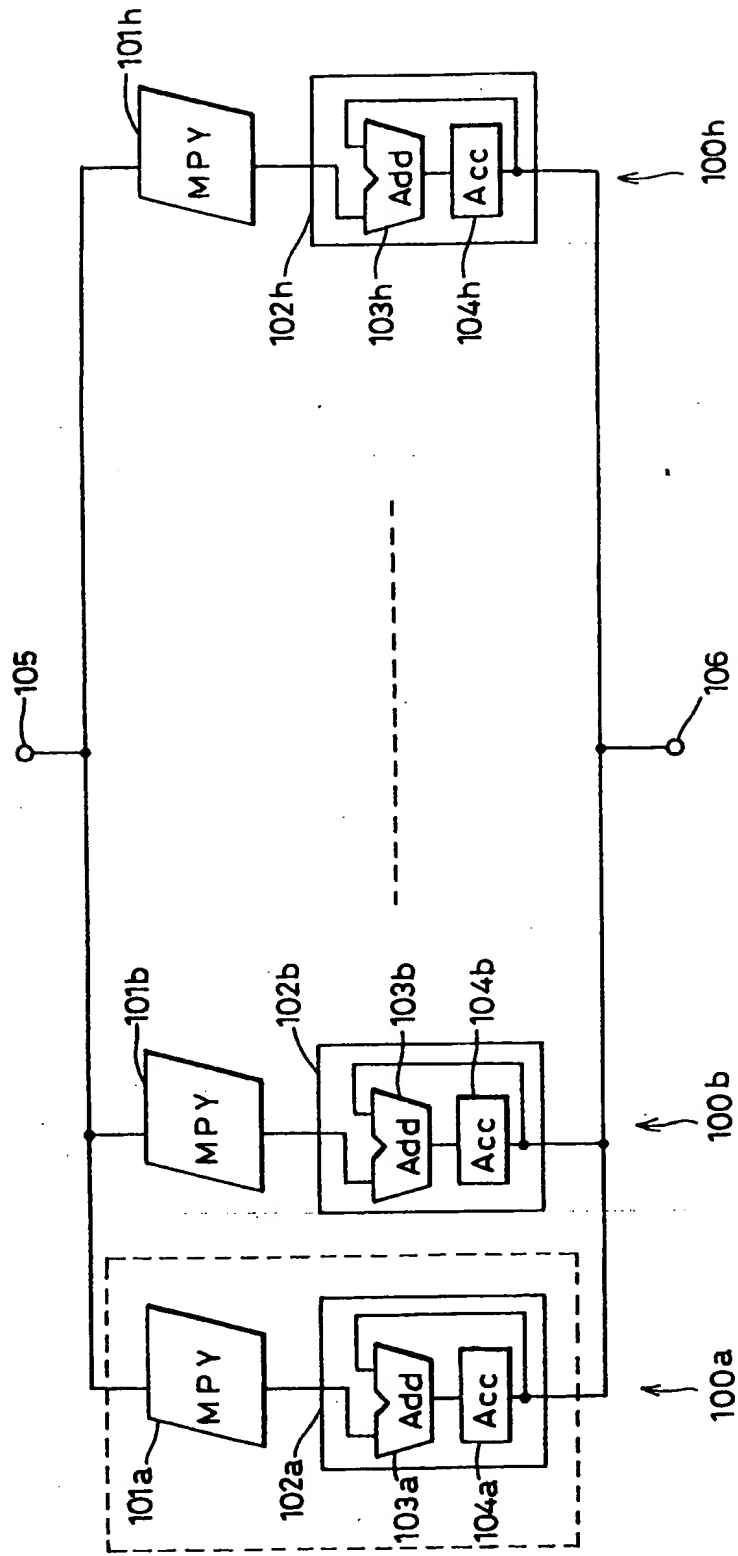


FIG. 2

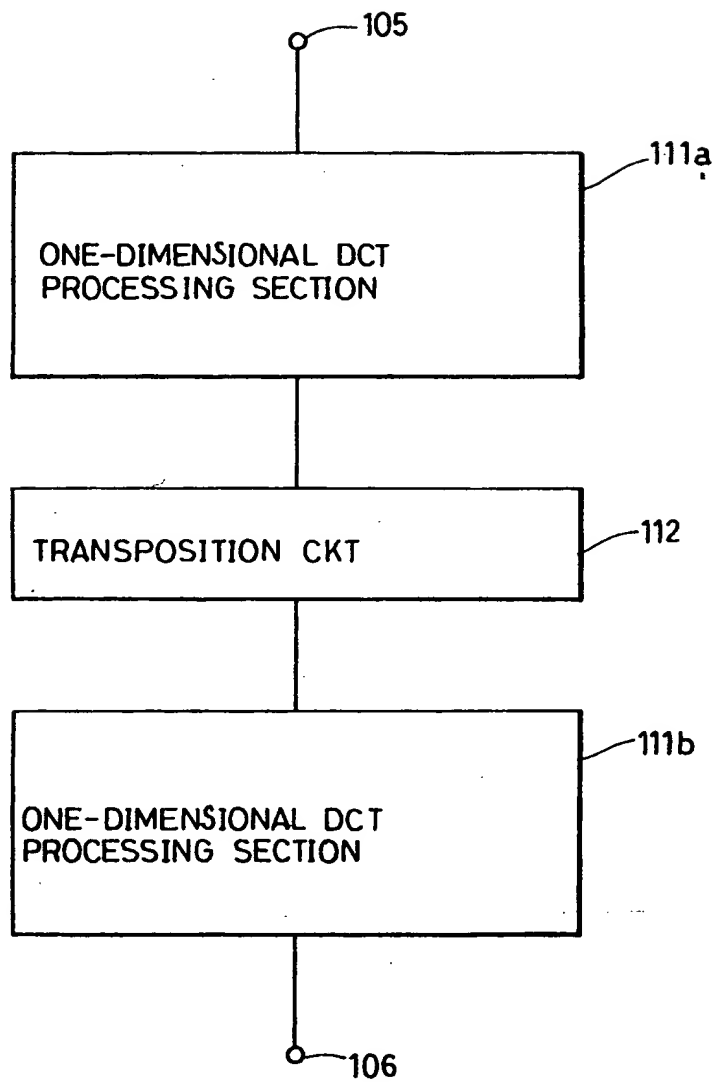


FIG. 3

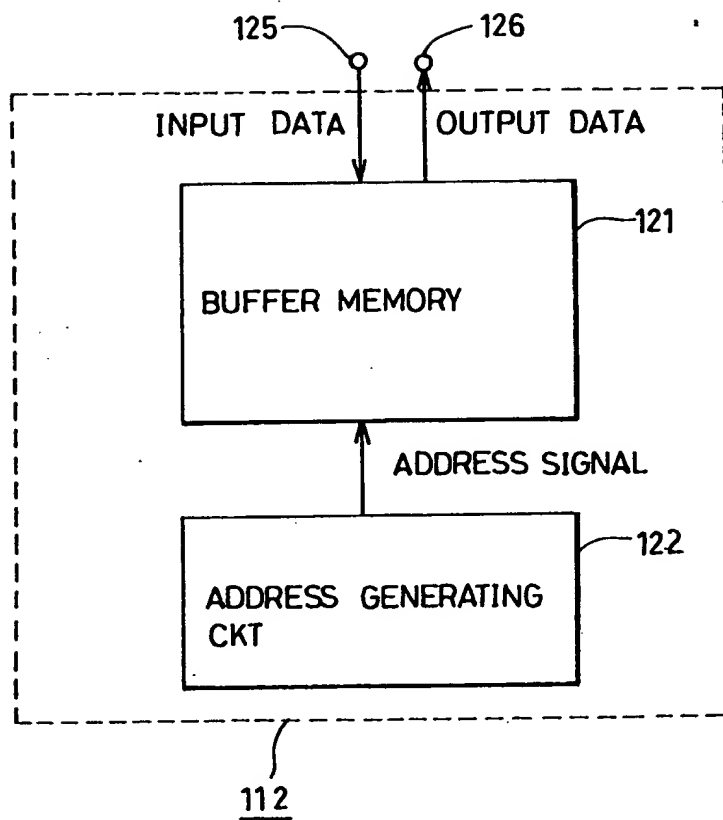


FIG. 4

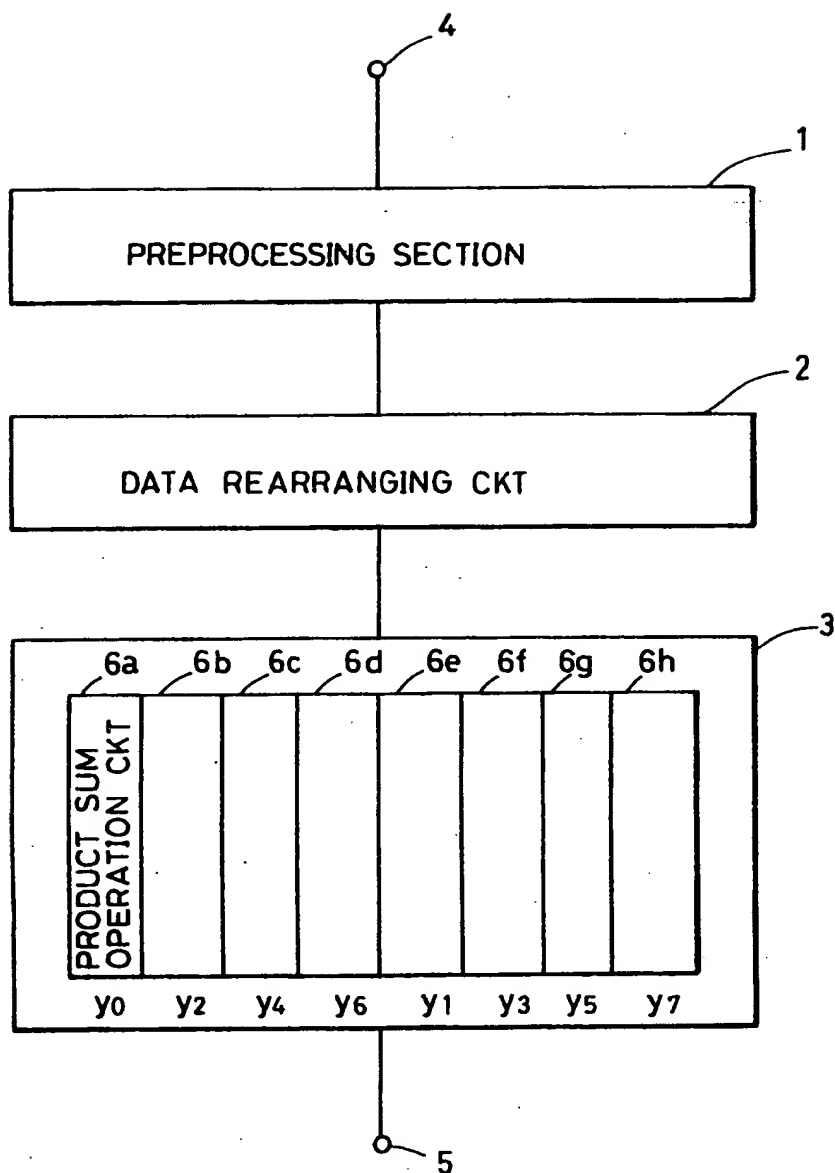


FIG. 5

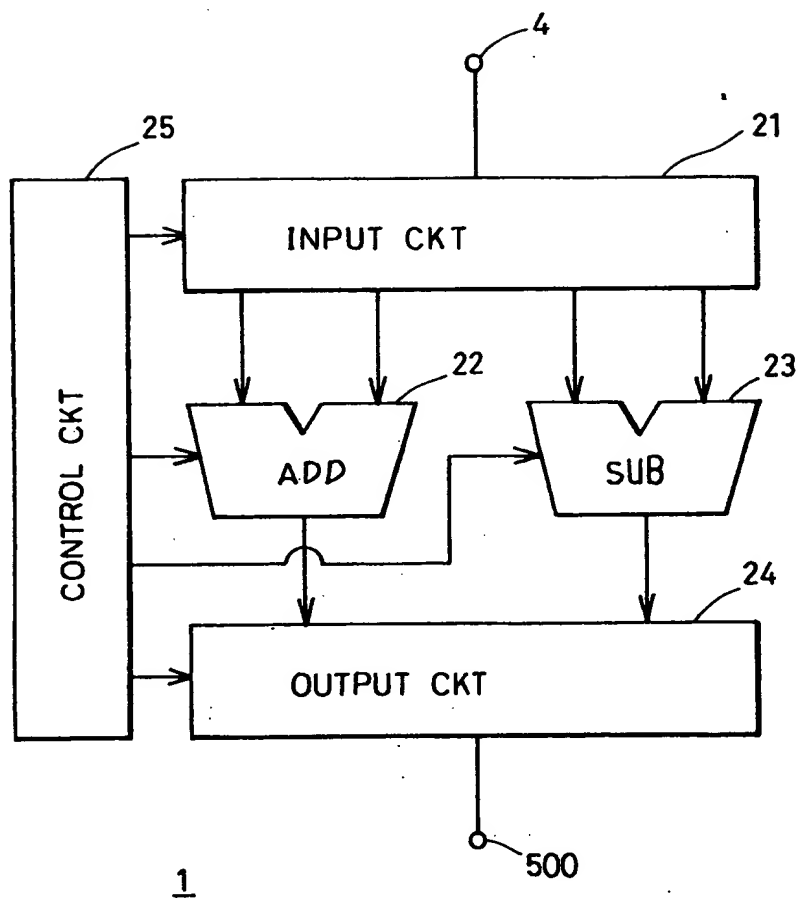
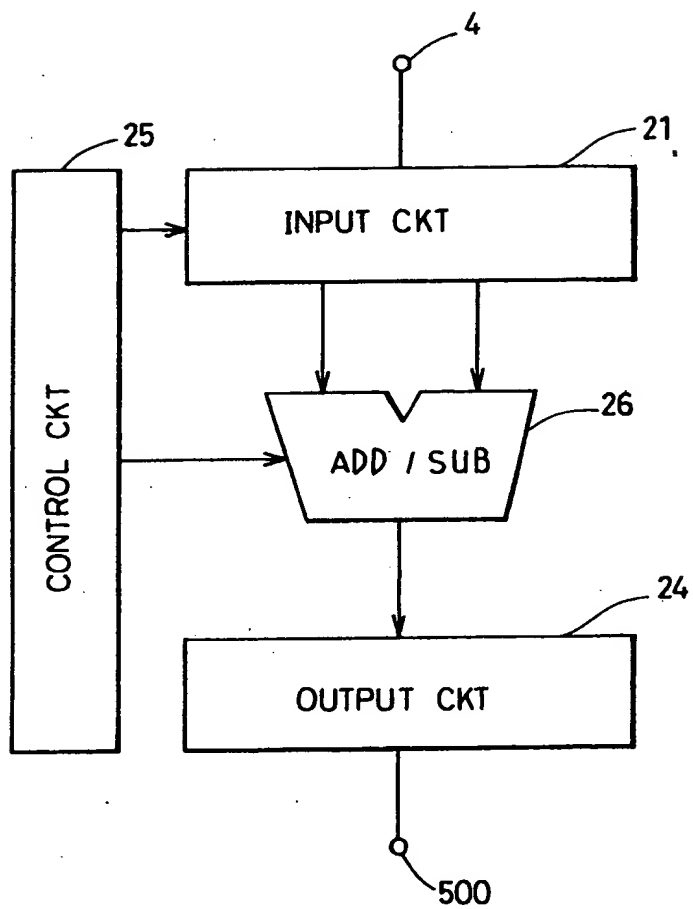


FIG.6



1

FIG. 7A

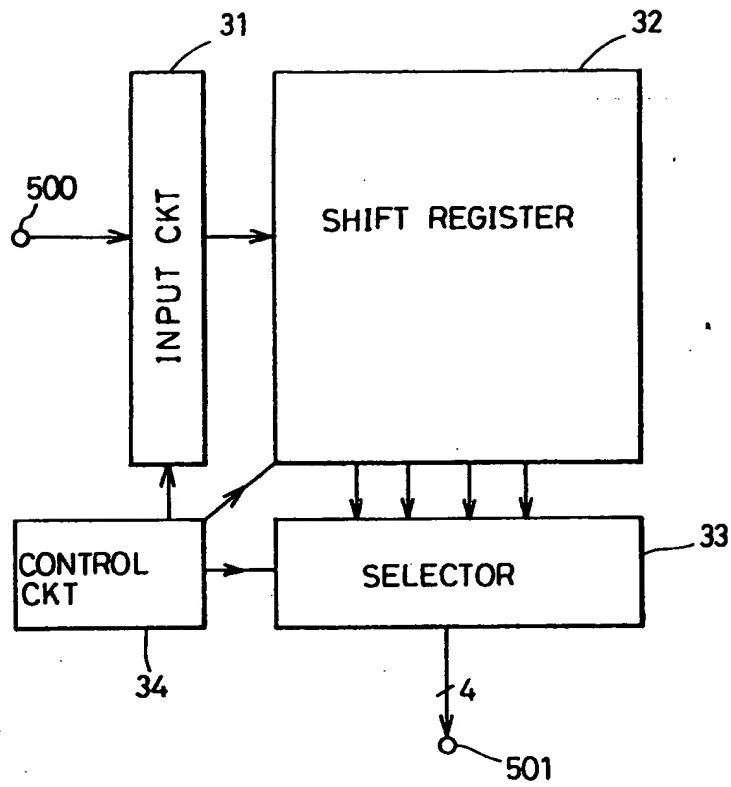


FIG. 7B

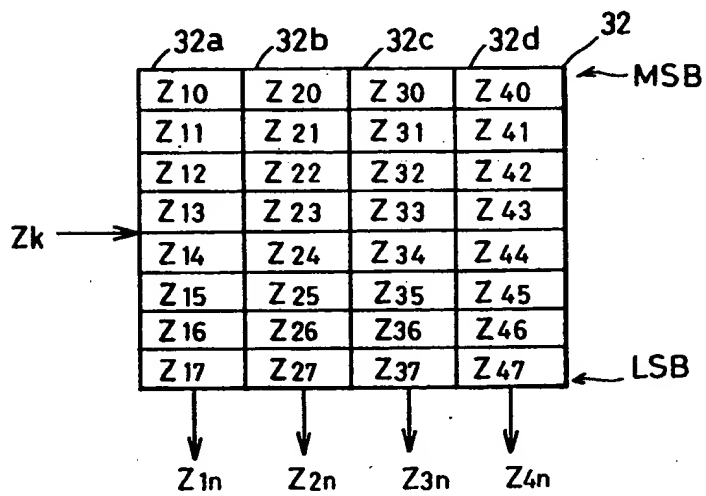
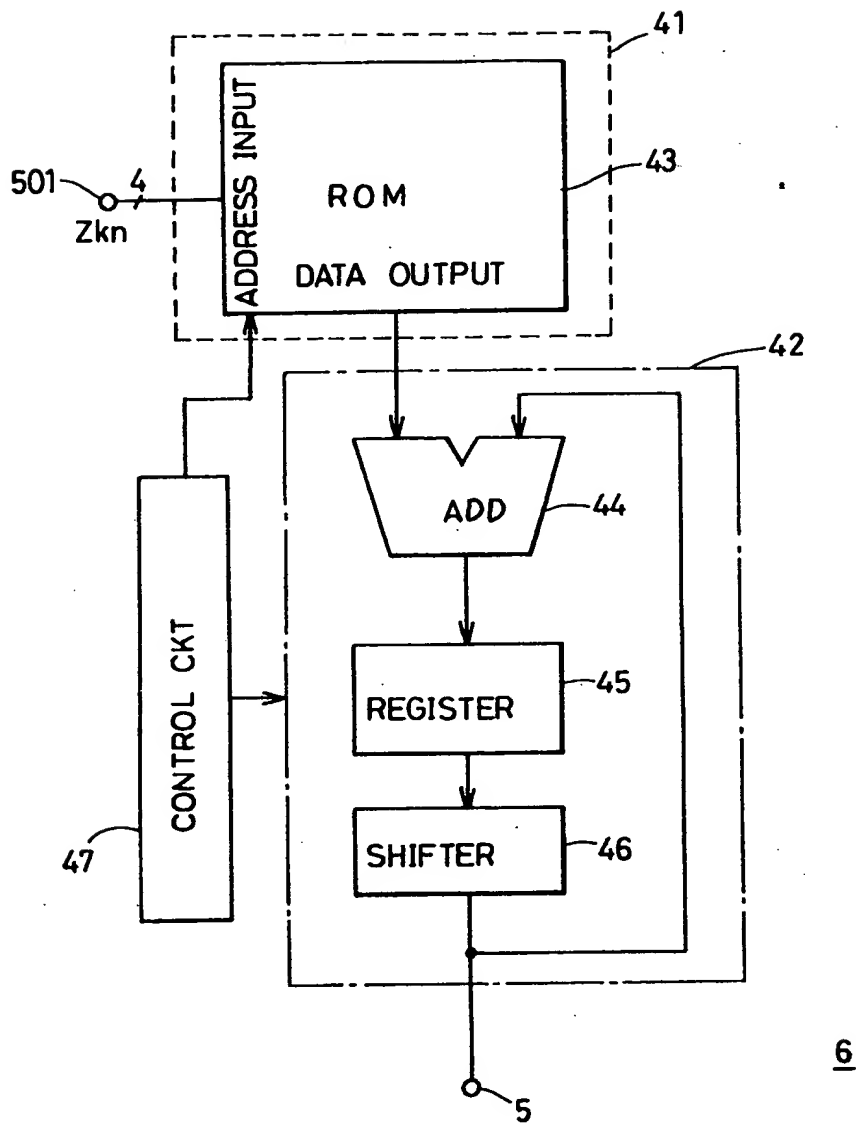


FIG. 8



6

FIG. 9

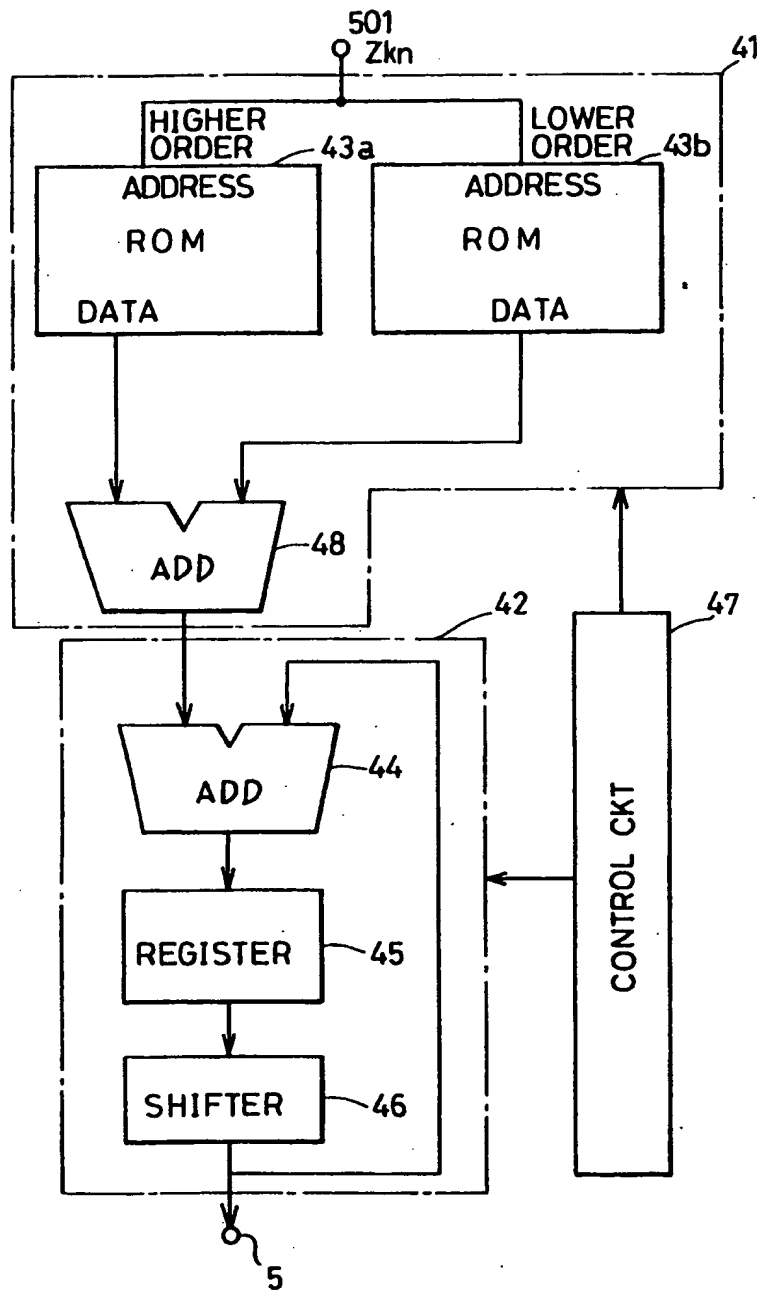


FIG.10

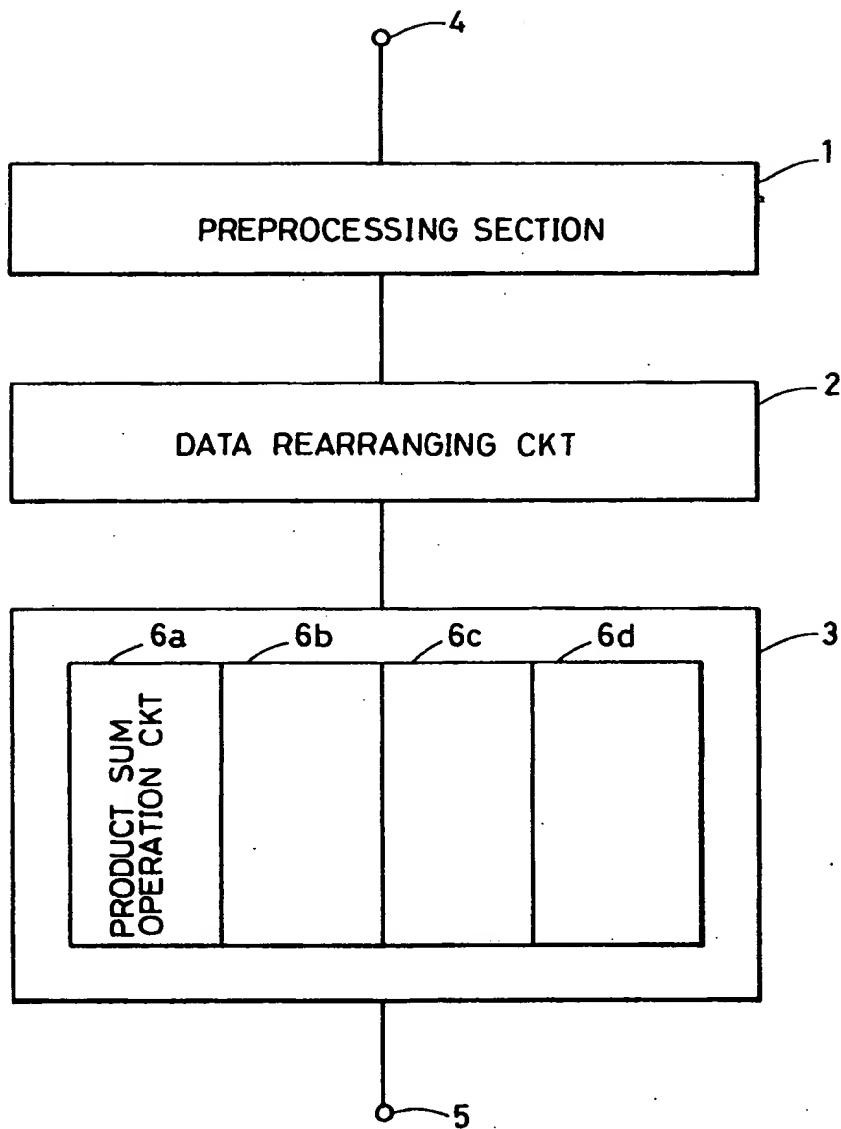


FIG. 11

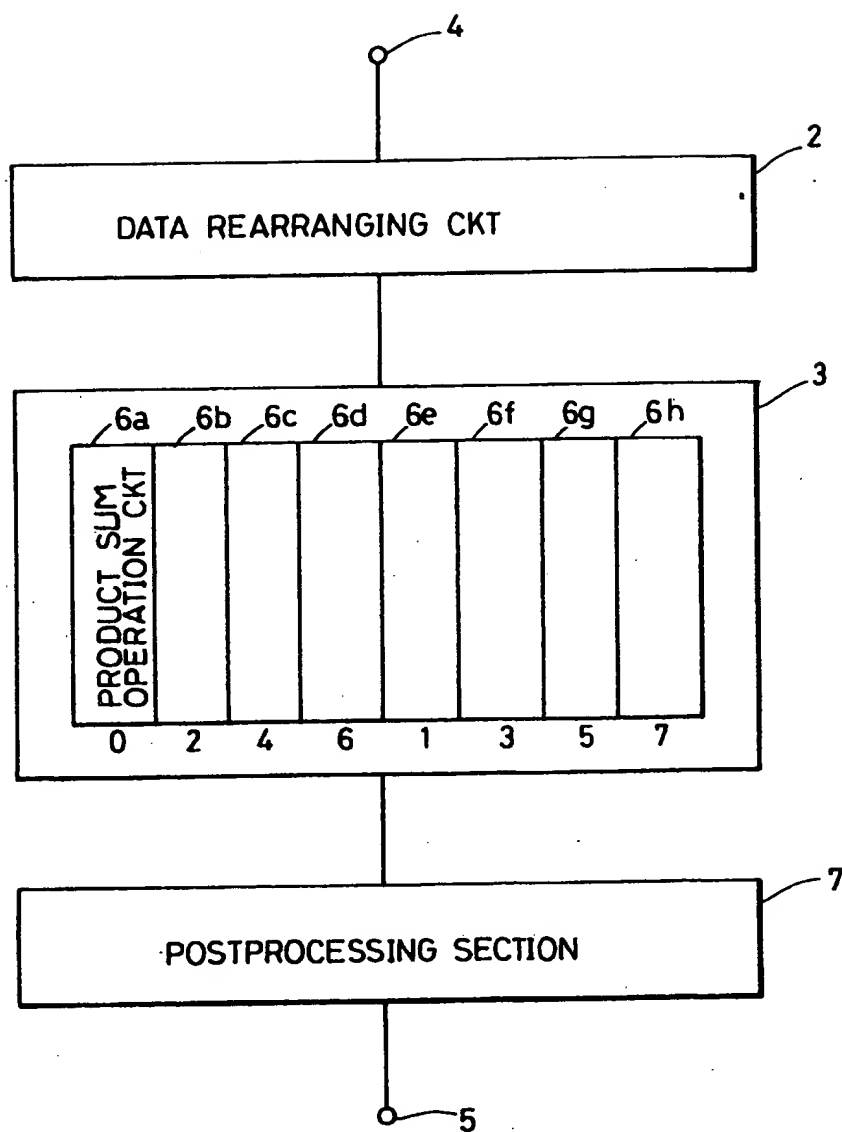


FIG. 12

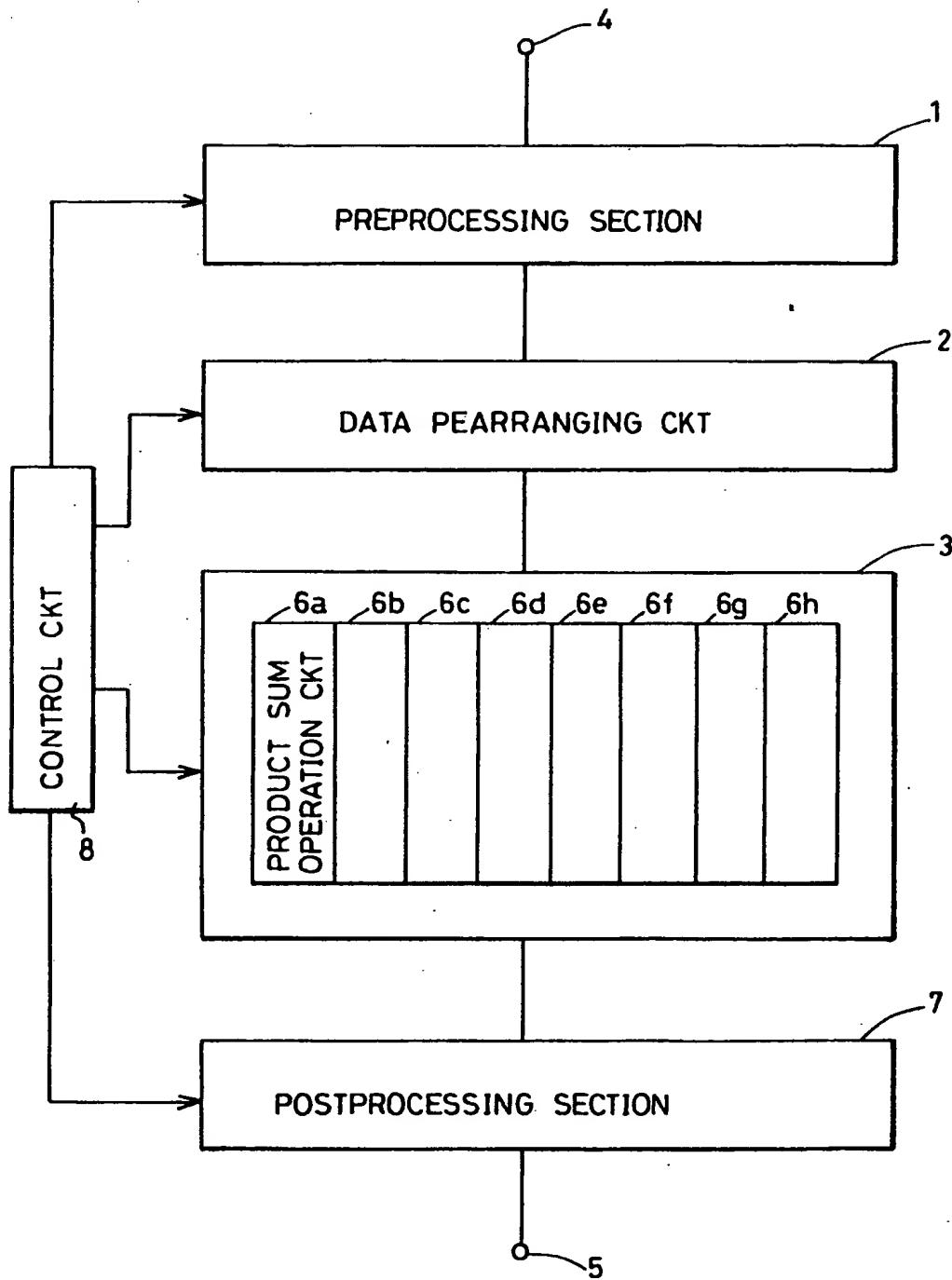


FIG. 13

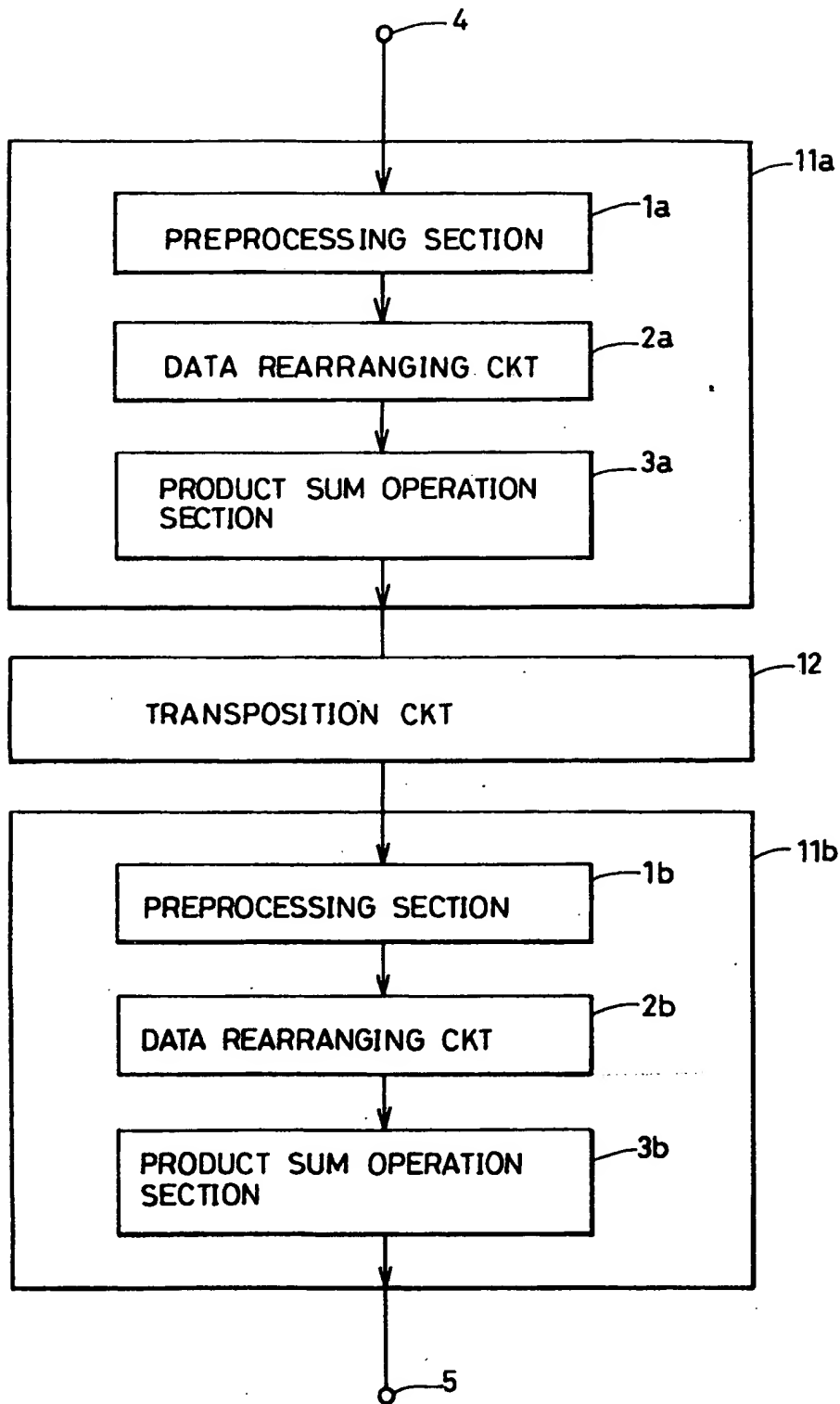


FIG. 14

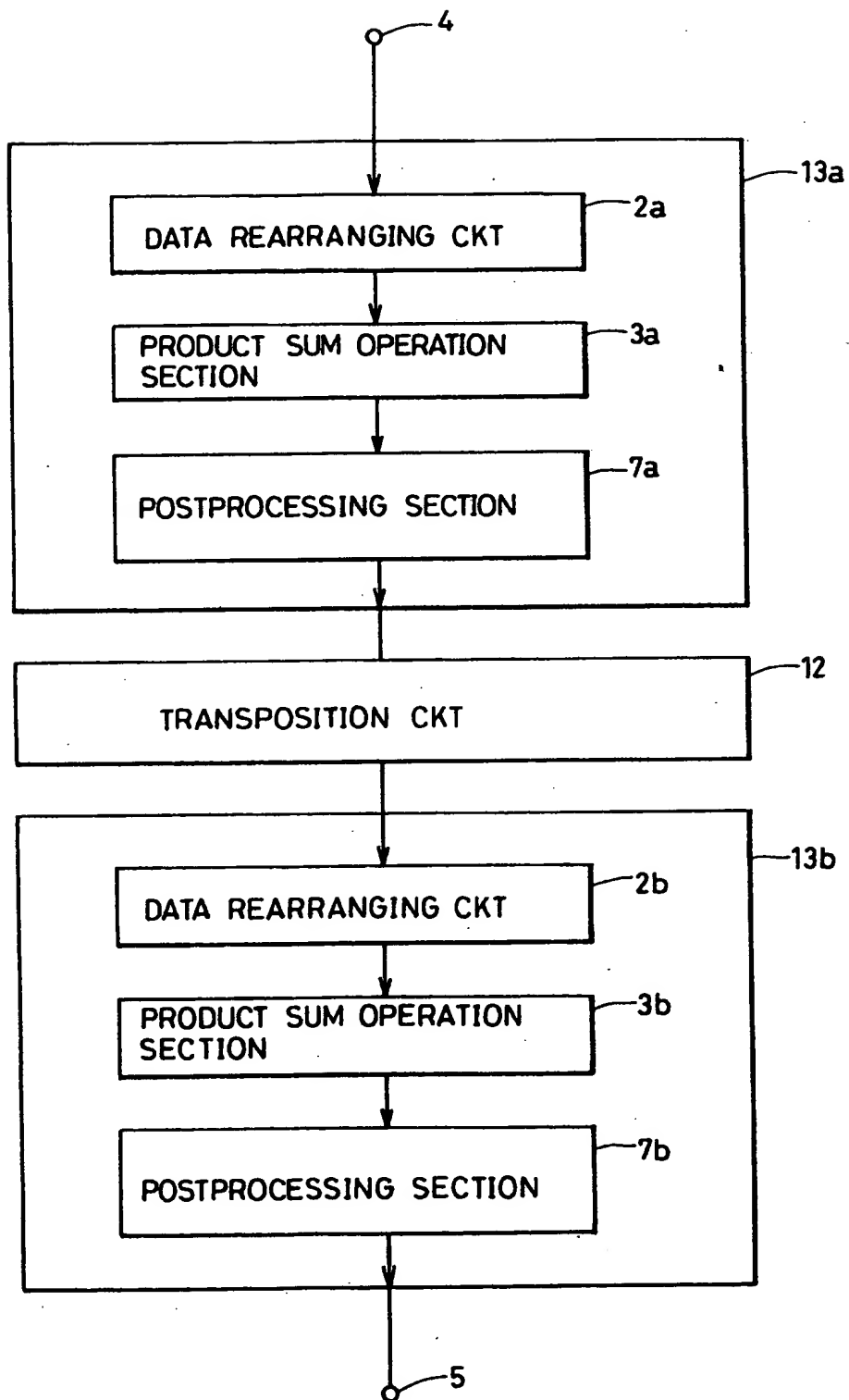


FIG. 15

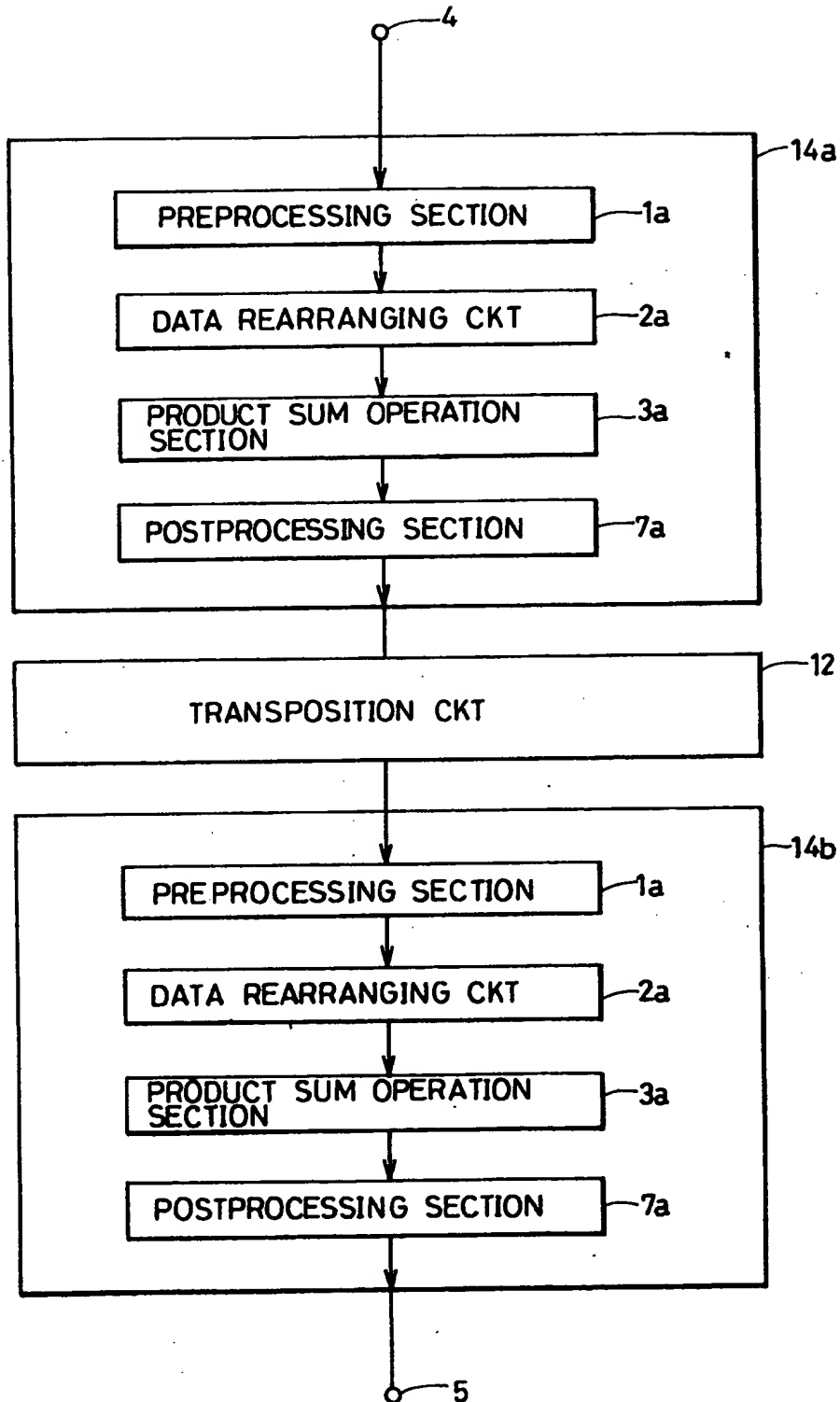


FIG.16

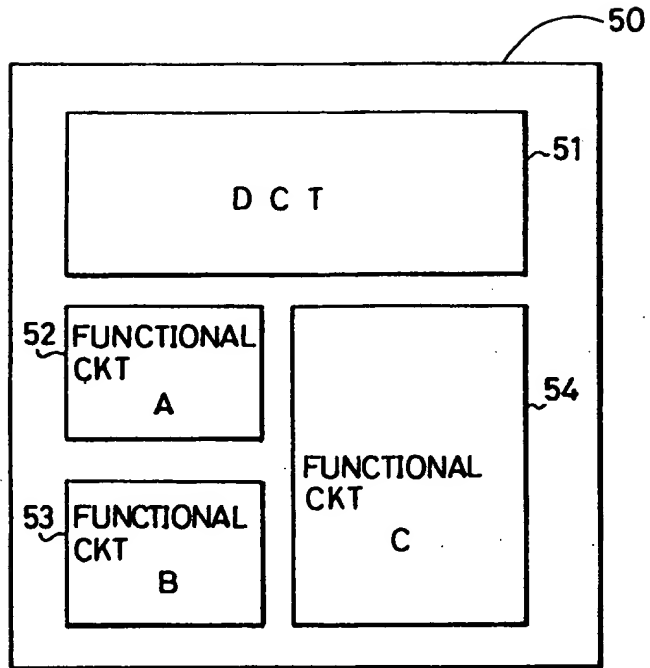
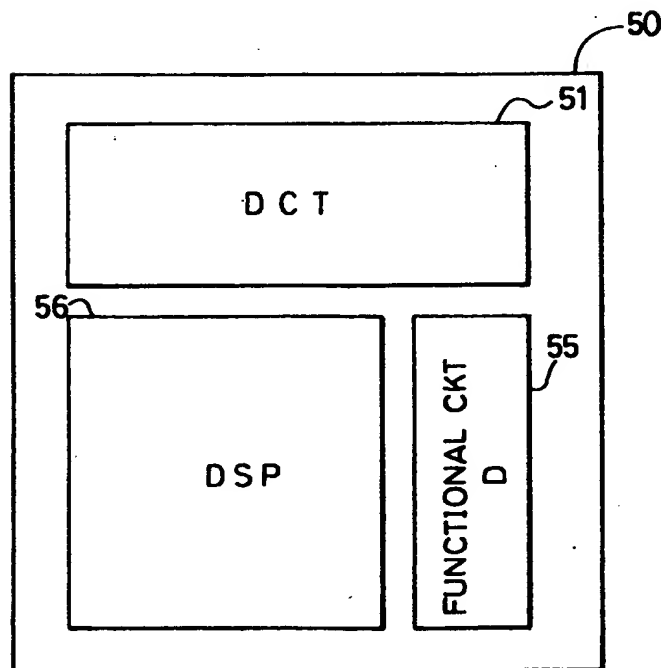


FIG.17



INTERNATIONAL
STANDARD

ISO/IEC
11172-3

First edition
1993-08-01

**Information technology — Coding of
moving pictures and associated audio for
digital storage media at up to about
1,5 Mbit/s —**

**Part 3:
Audio**

*Technologies de l'information — Codage de l'image animée et du son
associé pour les supports de stockage numérique jusqu'à environ
1,5 Mbit/s —*

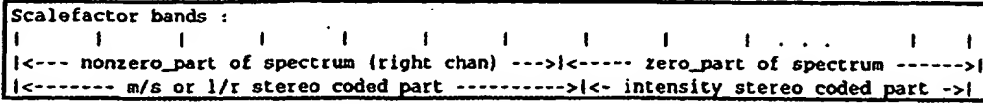
Partie 3: Audio



Reference number
ISO/IEC 11172-3:1993(E)

Best Available Copy

channel. Above this bound decoding of intensity stereo is applied using the scalefactors of the right channel as intensity stereo positions. An intensity stereo position of 7 in one scalefactor band indicates that this scalefactor band is not decoded as intensity stereo.



For each scalefactor band (sb) coded in intensity stereo, the following steps are executed:

- 1) the intensity stereo position is_pos_{sb} is read from the scalefactor of the right channel.
- 2) if ($is_pos_{sb} = 7$) do not perform the following steps (illegal is_pos).
- 3) $is_ratio = \tan\left(is_pos_{sb} \cdot \frac{\pi}{12}\right)$.
- 4) $L_i := L_i \cdot \frac{is_ratio}{1 + is_ratio}$ for all indices i within the actual scalefactor band sb .
- 5) $R_i := L_i \cdot \frac{1}{1 + is_ratio}$ for all indices i within the actual scalefactor band sb .

2.4.3.4.10 Synthesis filterbank

Figure A.4. shows a block diagram including the synthesis filterbank. The frequency lines are preprocessed by the "alias reduction" scheme (see the block diagrams in figure A.5 and in table B.9. for the coefficients) and fed into the IMDCT matrix, each 18 into one transform block. The first half of the output values are added to the stored overlap values from the last block. These values are new output values and are input values for the polyphase filterbank. The second half of the output values is stored for overlap with the next data granule. For every second subband of the polyphase filterbank every second input value is multiplied by -1 to correct for the frequency inversion of the polyphase filterbank.

2.4.3.4.10.1 Alias reduction

For long block_type granules (block_type != 2) the input to the synthesis filterbank is processed for alias reduction before processing by the IMDCT. The following pseudo code describes the alias reduction computation:

```

for (sb=1; sb<32; sb++)
  for (i=0; i<8; i++) {
    xar[18*sb-1-i] = xr[18*sb-1-i]Cs[i] - xr[18*sb+1]Ca[i]
    xar[18*sb+i] = xr[18*sb+i]Cs[i] + xr[18*sb-1-i]Ca[i]
  }

```

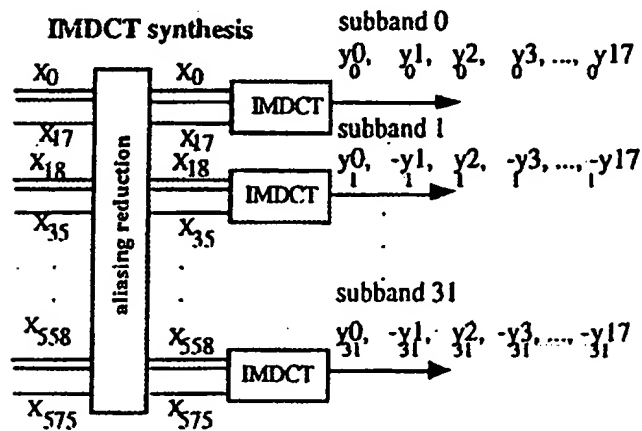
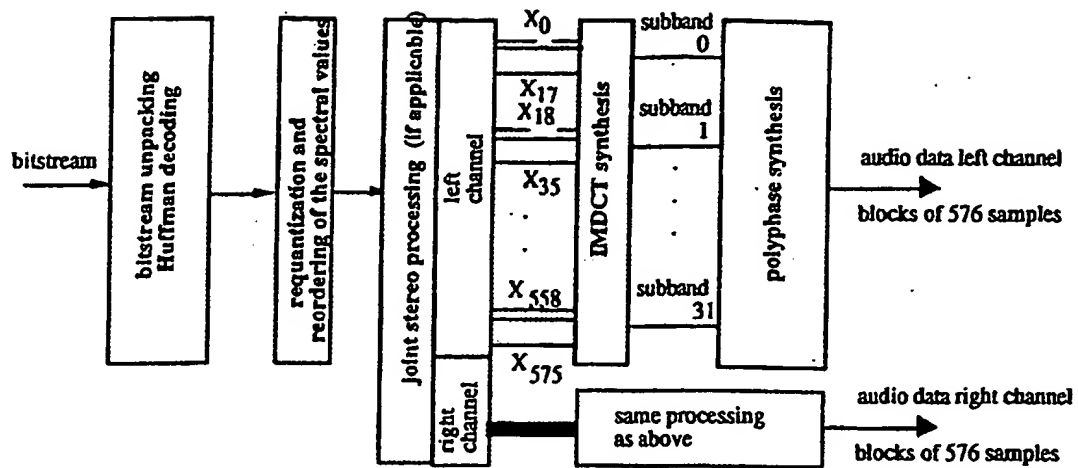
The indices of arrays xar[] and xr[] label the frequency lines in a granule, arranged in order of lowest frequency to highest frequency, with zero being the index of the lowest frequency line, and 575 being the index of the highest. The coefficients: Cs[i] and Ca[i] can be found in table B.9. Figures A.5 and A.6 illustrate the alias reduction computation.

Alias reduction is not applied for granules with block_type == 2 (short block).

2.4.3.4.10.2 IMDCT

In the following, n is the number of windowed samples (for short blocks n is 12, for long blocks n is 36). In the case of a block of type "short", each of the three short blocks is transformed separately. $n/2$ values X_k are transformed to n values x_i . The analytical expression of the IMDCT is:

$$x_i = \sum_{k=0}^{\frac{n}{2}-1} X_k \cos\left(\frac{\pi}{2n}\left(2i+1+\frac{\pi}{2}\right)(2k+1)\right) \quad \text{for } i = 0 \text{ to } n-1$$



Each IMDCT module calculates 18 output values $y_0..y_{17}$ out of 18 input spectral values. For every other subband every other output sample should be multiplied by -1, as shown in the diagram.

Figure A.4 -- Layer III decoder diagram

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.